

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

**VÍCETŘÍDÁ SEGMENTACE 3D LÉKAŘSKÝCH DAT  
POMOCÍ HLUBOKÉHO UČENÍ**

MULTICLASS SEGMENTATION OF 3D MEDICAL DATA USING DEEP LEARNING

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Tomáš Slunský**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Martin Kolařík**

**BRNO 2019**

# Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Tomáš Slunský

**ID:** 155870

**Ročník:** 2

**Akademický rok:** 2018/19

## NÁZEV TÉMATU:

### Vícetřídní segmentace 3D lékařských dat pomocí hlubokého učení

## POKyny PRO VYPRACOVÁNÍ:

Nastudujte současné metody segmentace obrazu, které používají tzv. hluboké učení. V rámci diplomové práce proveďte rešerši těchto metod a jejich vlastností se zaměřením na použití pro segmentaci dat do více tříd. Dále ze zkoumaných postupů vyberte vhodnou metodu a aplikujte ji po dohodě s vedoucím na vybraný dataset. Výběr zdůvodněte a metodu otestujte pro použití na vybraná data. Výsledky testování vhodně prezentujte.

## DOPORUČENÁ LITERATURA:

[1] LONG, Jonathan, Evan SHELHAMER a Trevor DARRELL, Fully convolutional networks for semantic segmentation [online]. [cit. 2017-09-13]. DOI: 10.1109/CVPR.2015.7298965. ISBN 10.1109/CVPR.2015.7298965. Dostupné z: <http://ieeexplore.ieee.org/document/7298965/>

[2] SCHMIDHUBER, Jürgen, Deep learning in neural networks: An overview [online]. [cit. 2017-09-13]. DOI: 10.1016/j.neunet.2014.09.003. ISBN 10.1016/j.neunet.2014.09.003. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0893608014002135>

**Termín zadání:** 1.2.2019

**Termín odevzdání:** 16.5.2019

**Vedoucí práce:** Ing. Martin Kolařík

**Konzultant:**

**prof. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## ABSTRAKT

Diplomová práce se zabývá segmentací obrazu do více tříd pomocí konvolučních neuronových sítí. Teoretická část práce se zaměřuje na souhrn problematiky segmentace obrazu, kde jsou popsány základní principy fungování jak samotných neuronových sítí, tak segmentací obrazu včetně popsání jejich různých typů a variant. V praktické části je zvolena pro segmentace obrazu a podrobněji popsána architektura U-net, která byla aplikována na dataset medicínských dat. Dále je tu popsán postup zpracování trojrozměrných dat, jejich předzpracování a metody, jakými byla provedena celá vícetřídá segmentace. Závěr práce vyhodnocuje dosažené výsledky a zasazuje je do širšího kontextu.

## KLÍČOVÁ SLOVA

hluboké učení, konvoluční neuronové sítě, segmentace obrazu, strojové učení, neuronová síť, umělá inteligence, vícetřídá segmentace obrazu

## ABSTRACT

Master's thesis deals with multiclass image segmentation using convolutional neural networks. The theoretical part of the Master's thesis focuses on image segmentation. There are basics principles of neural networks and image segmentation with more types of approaches. In practical part the U-net architecture is chosen and is described for image segmentation more. U-net was applied for medicine dataset. There is processing procedure which is more described for image processing of three-dimensional data. There are also methods for data preprocessing which were applied for image multiclass segmentation. Final part of current master's thesis evaluates results.

## KEYWORDS

deep learning, convolutional neural networks, image segmentation, machine learning, neural network, artificial intelligence, multi-class image segmentation

SLUNSKÝ, Tomáš. *Vícetřídá segmentace 3D lékařských dat pomocí hlubokého učení*. Brno, Rok, 68 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Martin Kolařík

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Vícetřídá segmentace 3D lékařských dat pomocí hlubokého učení“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych touto cestou poděkoval svému vedoucímu diplomové práce panu Ing. Martinu Kolaříkovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....  
podpis autora

# Obsah

<b>Úvod</b>	<b>12</b>
<b>1 Neuronové sítě</b>	<b>13</b>
1.1 Neuron . . . . .	14
1.1.1 Umělý neuron . . . . .	15
1.2 Princip fungování sítě . . . . .	16
1.3 Strojové učení . . . . .	17
1.3.1 Učení s učitelem . . . . .	17
1.3.2 Učení bez učitele . . . . .	18
1.4 Konvoluční neuronové sítě . . . . .	19
1.4.1 Konvoluční vrstva . . . . .	20
1.4.2 Sdružovací vrstva . . . . .	22
1.4.3 Plně propojená vrstva . . . . .	23
1.5 Dávková normalizace . . . . .	24
1.6 Aktivační funkce . . . . .	25
1.6.1 ReLU . . . . .	25
1.6.2 Sigmoid . . . . .	25
1.6.3 Tanh . . . . .	26
1.6.4 Softmax . . . . .	26
1.6.5 LReLU - Leaky ReLU . . . . .	27
<b>2 Analýza problému</b>	<b>28</b>
2.1 Segmentace obrazu . . . . .	28
2.2 Současné metody segmentace obrazu . . . . .	29
2.2.1 Detekce hran . . . . .	29
2.2.2 Prahování . . . . .	30
2.2.3 Metody založené na regionech . . . . .	32
2.2.4 Znalostní metody . . . . .	34
2.2.5 Hybridní metody . . . . .	34
2.3 U-net . . . . .	37
2.3.1 Architektura . . . . .	37
2.4 Vývojové prostředí . . . . .	39
2.4.1 Tensorflow . . . . .	39
2.4.2 Keras . . . . .	39
2.4.3 Numpy . . . . .	39
2.4.4 Scikit . . . . .	40
2.5 Vstupní data . . . . .	41



2.5.1	Magnetická rezonance - MRI . . . . .	41
2.5.2	Popis zpracovávaných dat . . . . .	41
<b>3</b>	<b>Implementace vícetřídé segmentace</b>	<b>45</b>
3.1	Transformace vstupních dat . . . . .	45
3.1.1	Zpracování vstupních dat a proces výpočtů . . . . .	46
3.1.2	Základní zpracování dat a načtení do numpy . . . . .	46
3.2	Extrakce 3D obrázku na menší části . . . . .	47
3.2.1	Rozdělení 3D skenu pomocí posuvného 3D okna . . . . .	48
3.3	Augmentace dat . . . . .	48
3.3.1	Aplikace augmentace . . . . .	49
3.4	Vytvoření modelu a jeho parametry . . . . .	50
3.5	Rekonstrukce dat . . . . .	50
<b>4</b>	<b>Výsledky a jejich vyhodnocení</b>	<b>52</b>
4.1	Metriky pro hodnocení segmentace . . . . .	52
4.1.1	Dice koeficient . . . . .	52
4.1.2	Hausdorffova vzdálenost . . . . .	52
4.1.3	Snímková podobnost . . . . .	53
4.2	Náročnost výpočtu a parametry trénování . . . . .	53
4.3	Průběh trénování neuronové sítě . . . . .	53
4.3.1	Trénování pomocí celých 3D skenů . . . . .	54
4.3.2	Trénování pomocí fragmentů . . . . .	55
4.4	Výsledky segmentace . . . . .	56
4.4.1	Přesnost trénování po celých 3D skenech . . . . .	56
4.4.2	Přesnost trénování po fragmentech . . . . .	57
4.4.3	Výsledná 3D segmentace . . . . .	58
4.5	Vyhodnocení výsledků a možné pokračování práce . . . . .	60
<b>5</b>	<b>Závěr</b>	<b>61</b>
	<b>Literatura</b>	<b>62</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>66</b>
	<b>Seznam příloh</b>	<b>67</b>
<b>A</b>	<b>Obsah přiloženého CD</b>	<b>68</b>

# Seznam obrázků

1.1	Biologický neuron.[1]	14
1.2	Umělý neuron	15
1.3	Neuronová síť.	16
1.4	Učení bez učitele - roztrídění vstupních dat do tříd.	18
1.5	Architektura konvoluční neuronové sítě.	19
1.6	2D konvoluce.	20
1.7	Max pooling.	22
1.8	ReLU	25
1.9	Sigmoid	26
1.10	Tanh	26
1.11	LReLU	27
2.1	Jednotlivé typy hran.	30
2.2	Prahování.	31
2.3	K-means.	33
2.4	Architektura U-net. Převzato z [2]	37
2.5	2D obrázek řezu mozku.	43
2.6	3D obrázek řezů mozku.	44
3.1	3D patche [3]	47
3.2	3D patche	48
3.3	Ukázka sestavení jednoho řezu z fragmentů.	51
4.1	Průběh trénování po celých skenech.	54
4.2	Průběh trénování.	55
4.3	Řez výsledné vícetřídní segmentace.	57
4.4	Výsledná 3D segmentace celého mozku.	58
4.5	Výsledná 3D segmentace mozku do více tříd.	59

# Seznam tabulek

2.1	Úplná tabulka segmentovaných tříd.[4]	42
4.1	Výsledky trénování pomocí celých skenů.	56
4.2	Výsledky trénování pomocí fragmentů.	57

## Seznam výpisů

# Úvod

V dnešní době se čím dál více prosazuje pro zpracování obrazu umělá inteligence resp. neuronové sítě. Důvodem je ten, že neuronové sítě jsou postaveny na základech, které věrně přebírají vzory z přírody a aplikují je v informatice. Využíváním takových procesů se ukázalo, že návrh na základě neuronových sítí dokáže některé problémy řešit daleko efektivněji nežli současně používané metody. Jedním z těchto problémů je segmentace obrazu pomocí neuronových sítí, kdy dochází k rozdělení daného obrazu na oblasti se společnými vlastnostmi. Tyto vlastnosti pak mají určitý význam pro celý obrázek, z kterého byly vysegmentovány.

Jelikož počítač nedokáže rozlišit obraz tak, jak jej dokáže zpracovávat náš mozek, je důležité takový obraz, např. pořízený pomocí MRI (Magnetická rezonance - *Magnetic resonance imaging*), dále zpracovávat - segmentovat. Výstupní vysegmentovaný obraz je poté lépe využitelný v dalším zpracování nebo analýze.

Následující práce se zabývá segmentací obrazu do více tříd, kdy dochází k rozlišení více než jedné částí se společnými vlastnostmi a následně k jejich vyhodnocení. V diplomové práci se vícetřídní segmentace využívá k rozlišení jednotlivých částí mozku. Takový postup je pak dále velice přínosný pro další zpracování, například pro diagnostiku určitých onemocnění mozku apod.

Mezi hlavní přínosy práce patří segmentace mozku do více tříd, kdy neuronová síť bude schopná jednotlivé části mozku rozlišit ze vstupních dat MRI, čímž by práce mohla v budoucnu přispět k minimalizaci chyb při diagnostice mozkových chorob a k vyšší rychlosti analýzy daného problému.

Obsahová část diplomové práce je členěna do čtyř hlavních částí. První část se věnuje obecnému úvodu do neuronových sítí, jejich rozdělení, architektuře apod. Druhá část práce se zaměřuje podrobněji na segmentaci obrazu, dále pojednává o zvolené metodě U-net pro segmentaci a nakonec shrnuje vývojové prostředí, které bylo využíváno během trénování sítě. Předposlední část se věnuje analýze hlavního cíle práce, jejího návrhu a implementaci. Poslední část prezentuje dosažené výsledky, návrhy na zlepšení a možné pokračování.

# 1 Neuronové sítě

Počátek neuronových sítí sahá až do čtyřicátých let minulého století, kdy byly položeny základy pro vznik oborou neuronových sítí. Neuronové sítě jako takové, jsou modelem biologických struktur tak, jak je známe dnes v živých organismech.[5][6] Pro účely modelu proto byl vytvořen tzv. umělý neuron, což je matematický popis biologického neuronu. Skládá se z množin vstupů a jednoho výstupu. O tom jaký vstup se dostane na výstup pak rozhodují váhy a funkce. Neuronová síť je poté tvořena obrovskou množinou takto definovaných neuronů. Jednou z největších předností takových sítí je proces učení, tedy proces, kdy po jednotlivých průchodech neuronovou sítí jsou upravovány váhy. Tento proces vede k tomu, že neuronová síť umožňuje lépe manipulovat se vstupními daty.[5]

Největší přínos neuronových sítí tkví v situacích, kdy není možné použít matematický popis všech souvislostí a vztahů daného problému, nebo v případech, kdy by po sestavení takového modelu byly veškeré operace nad ním příliš složité až nemožné. Doménou neuronových sítí se stávají úlohy pro zpracování, rozpoznávání, optimalizaci nad daty a informacemi, které jsou nepřesné, neúplné, neurčité nebo není jiná metoda, jak by deterministický systém byl schopen takové informaci pochopit tj. zpracovat. [7]

Jedním z důvodů tak obrovského nárůstu významnosti neuronových sítí je ten, že jsou schopny zareagovat na **neznámé** vstupy, na které nebyly doposud trénovány, jedná se tedy o schopnost zobecňování získaných poznatků na základě předešlých zkušeností.[8][7]

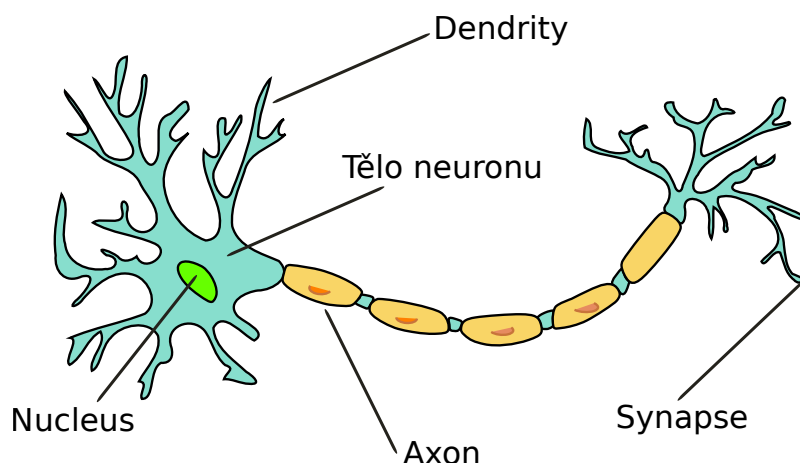
## Základní uspořádání neuronových sítí

- **Dopředná neuronová síť** – hrany jsou orientovány jedním směrem zleva doprava, signál se tedy již nešíří zpět. Výstupy jedné vrstvy neuronu jdou na vstup druhé vrstvy až po výstupní vrstvu, která reprezentuje konečný výstup celé neuronové sítě.
- **Rekurentní neuronová síť** – Hlavním rozdílem oproti dopředným sítím je ten, že neurony mohou být ještě propojeny zpětně tj. zpětnou vazbou popřípadě mohou obsahovat smyčky zpět k určitému neuronu, což je vhodné k řešení iterativních úloh.

## 1.1 Neuron

Jak již bylo nastíněno výše, základní stavební jednotkou umělých neuronových sítí je **neuron**. Jejich návrh je inspirován biologickou stavbou neuronu viz obr.1.1.[9]

Biologický neuron se skládá z těla neuronu, jádra (nucleus), dále dlouhých výběžků, které jsou nazývány také jako axony a krátkých výběžků (dendrity). Jednotlivé neurony jsou v mozku spojovány pomocí synapsí, které slouží k předávání vzruchů.



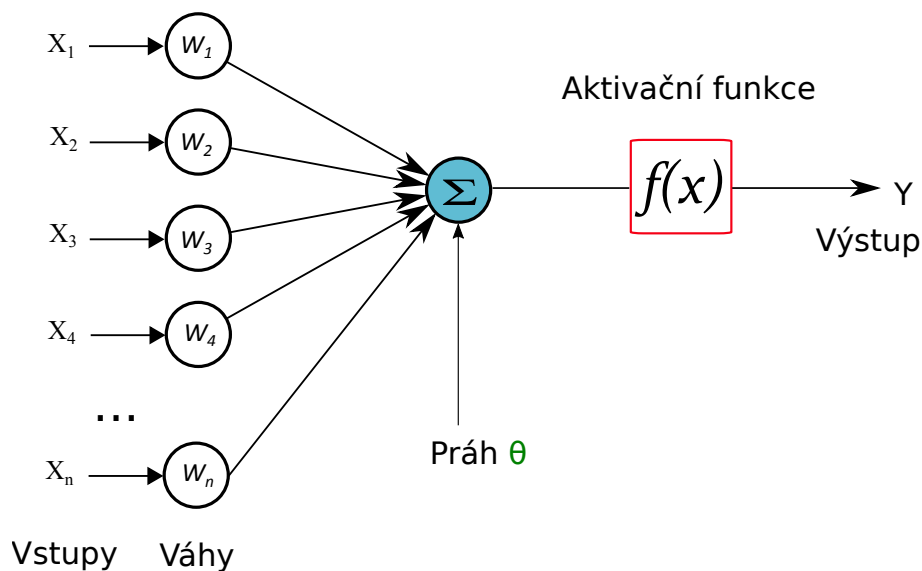
Obr. 1.1: Biologický neuron.[1]

Dendrity slouží v případě biologického neuronu k získávání vstupních informací (vzruchů) do těla neuronu od okolních neuronů. Axon slouží k šíření elektrochemických signálů na výstup neuronu. Šíření signálu po axonu je možné díky změnám elektrického náboje na jeho těle.[10] Na jeho konci opět dochází k „napojení“ pomocí synapsí na dendrity dalších neuronu atp.[11]

Termín „napojení“ by v tomto případě měl být více rozveden, jelikož neurony spojené ve skutečnosti nejsou, jsou mezi nimi mezery, takže elektrický vzruch by zde nemohl být fyzicky veden.[10] Synapse disponují tzv. neurotransmitery, které se navazují na receptory průchodem přes membránu. Signál na konci axonu způsobí, že neurotransmitery překonají štěrbinu, která je mezi dvojící membrán neuronů a naváží se na již výše zmiňovaný receptor. Tím je proces přenosu úplný.

### 1.1.1 Umělý neuron

Návrh umělého neuronu vychází ze svého biologického vzoru, ovšem bylo nutné aplikovat zjednodušení, aby bylo možné vytvořit takový matematický model. Dalo by se tedy říci, že matematická obdoba se vyvíjela poté již nezávisle vůči své původní předloze.[12] Vstupní informace jsou reprezentovány množinou hodnot  $X$  (modelují dendrity u biologického neuronu), které jsou váhově ohodnoceny a po překonání určité prahové hodnoty slouží jako vstup aktivační funkce  $f(x)$ . [13]



Obr. 1.2: Umělý neuron

Jednotlivými váhami u vstupů je v neuronu modelováno učení neboli uložení zkušeností ze zpracování předchozích průchodů. Čím je váha vstupu pak vyšší, bude i vstup neuronu důležitější a tedy preferovanější viz. obr.1.2. Výstupní hodnota neuronu je pak dána aktivační funkcí  $f(x)$  a zaslána na výstup  $Y$  po překonání prahu  $\theta$ . Aktivační funkce by měla být nelineární. Umělý neuron lze definovat matematickým zápisem:

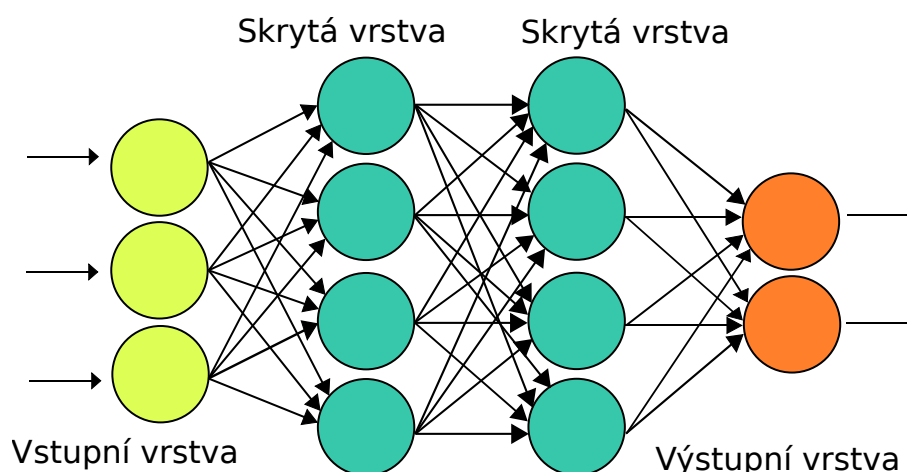
$$Y = f \left( \sum_{i=1}^n (W_i \cdot X_i) + \theta \right) \quad (1.1)$$



## 1.2 Princip fungování sítě

Propojením více neuronů vzniká neuronová síť, kdy výstup jednoho neuronu je vstupem jednoho či více dalších neuronů. Vzájemné propojení neuronů a jejich počet vede k vytvoření architektury na základě vrstev. V architektuře rozlišujeme tři typy vrstev. Vstupní vrstva neuronu je vrstva, která z vnějšího prostředí lze vidět, tato vrstva je dále spojena se skrytou vrstvou, kterou již z vnějšího prostředí vidět nelze. Skrytých vrstev může být rámci sítě větší množství. Skryté vrstvy jsou pak napojeny na ty výstupní. V rámci jedné vrstvy ovšem neurony propojeny nejsou.

Napříč různými vrstvami je propojení neuronů tzv. **úplné**. Změnou stavu jednotlivých neuronů, které leží na cestě mezi vstupními a výstupními neurony, je umožněno šíření a zpracování informace. Konfigurace sítě je poté reprezentována váhami. Největší dopad na stav sítě má proces učení, kdy síť upravuje svoje hodnoty na základě zpětné vazby po průchodu sítí (v případě učení s učitelem) popřípadě reklasifikace dat ve třídách (v případě učení bez učitele).



Obr. 1.3: Neuronová síť.

Na obrázku výše 1.3 lze vidět typickou dopřednou neuronovou síť. Neurony v rámci jedné vrstvy propojené nejsou tj. informace se šíří pouze zleva doprava. Situace by byla jiná v případě rekurentní neuronové sítě již zmiňované výše.

## 1.3 Strojové učení

Jedním z nezajímavějších aspektů neuronových sítí je bezpochyby jejich učení. Před příchodem neuronových sítí bylo běžnou praxí, že vstupní data se transformovaly na ty výstupní pomocí určitého systému zpracování neboli algoritmu. Při použití neuronových sítí je ale tato fáze plně zautomatizována pomocí učení, což je proces, při kterém dojde k natrénování sítě na určitý problém a ta je schopna jej pak řešit v situaci, kdy se ji předloží data, na kterých nikdy netrénovala, neboli fáze **vybavování**. Rozlišují se dvě fáze učení - s učitelem a bez učitele.

### 1.3.1 Učení s učitelem

Učení s učitelem je charakteristické tím, že k určité trénovací množině, která bude mít vliv na nastavení sítě, existuje správná výstupní množina dat tj. referenční hodnoty, podle kterých neuronová síť kalibruje parametry tak, aby se výstupní data sítě co nejvíce podobala těm referenčním. Učitelem jsou tedy v takovém případě myšlena správná výstupní data.

Jak již bylo nastíněno výše, celé učení neuronových sítí závisí na nastavení vah u jednotlivých neuronů, které jsou zase závislé na nastavení prahu a zvolené aktivační funkci. Pro úpravu vah je zaveden koeficient rychlost učení, kterým se definuje velikost kroku při jejich úpravě. Tato velikost kroku má poté vliv na to, jak rychle se nastavené váhy v čase mění.

Přesnost výstupů sítě se pak odvíjí od počtu trénovacích cyklů, při kterých docházelo k přenastavování vah. Těmto trénovacím cyklům se říká *epochy* a s rostoucím počtem epoch je i neuronová síť výkonnější. Obvykle je k naučení sítě potřeba větší počet epoch.

Epochy ale nejsou jediné, které mají vliv na přesnost neuronové sítě. Obrovský vliv má též velikost trénovací množiny, díky které si síť rozšiřuje svoje zkušenosti.

Ke zjištění přesnosti neboli výkonnosti sítě slouží testovací/validační množina dat. Jedná se cíleně o data, s kterými se neuronová síť ještě nesetkala (na kterých netrénovala), aby se zjistila její odezva resp. přesnost. Ke zjištění přesnosti se využívají různá kritéria, nejčastěji se používá střední aritmetická odchylka aplikovaná na výstupní data sítě s referenčními hodnotami.

Aby síť mohla upravit svoje hodnoty na základě zjištěné chyby, musí použít některý z algoritmů šíření chyby. Tím nejrozšířenějším pro tento účel se používá *backpropagation* neboli algoritmus zpětného šíření chyby. Jeho funkce spočívá v tom, že určí chybu neuronů na výstupech výstupní vrstvy a šíří ji zpět na vrstvy předcházející. Algoritmus má tyto tři etapy[5]:

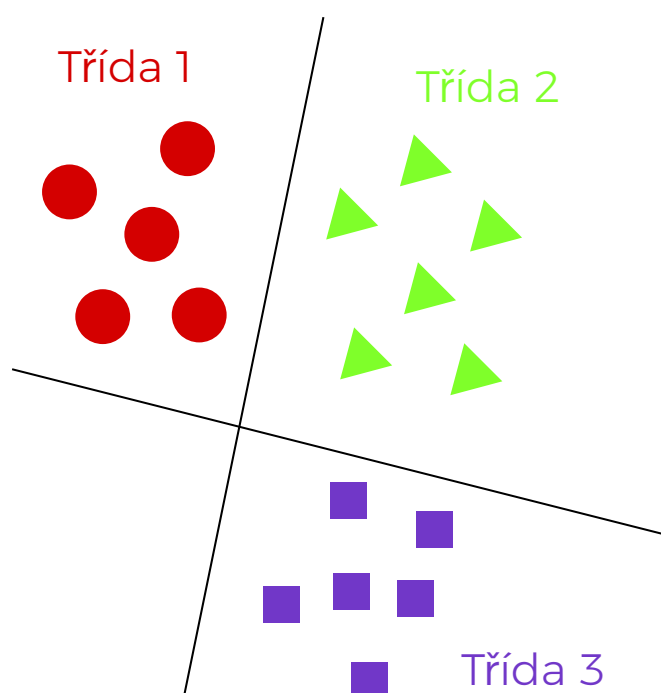
- Dopředné šíření vstupního signálu.
- Zpětné šíření chyby.
- Aktualizace váhových hodnot na jednotlivých spojeních.

### 1.3.2 Učení bez učitele

Metoda učení bez učitele nemá k dispozici množinu výstupních hodnot, tudíž není schopna ověřovat svoje výstupy způsobem, jako předcházející metoda. Veškeré informace, kterými disponuje, jsou pouze vstupní data resp. množina trénovacích dat.

Neuronová síť tedy v takovém případě začne data třídit do skupin dle vzájemné podobnosti tj. objekty o stejných či podobných vlastnostech[11].

Změnou vah se pak síť snaží dosáhnout konzistentního výstupu tj. pokud jsou na vstup přivedena data stejná či podobná o určitých vlastnostech, neuronová síť bude schopná tyto data správně zařadit a poskytnou odpovídající výstup<sup>1</sup> s ohledem na data, na kterých byla trénována.



Obr. 1.4: Učení bez učitele - roztrídění vstupních dat do tříd.

---

<sup>1</sup>Na stejné či podobné vstupní hodnoty bude reagovat adekvátním výstupem.

## 1.4 Konvoluční neuronové sítě

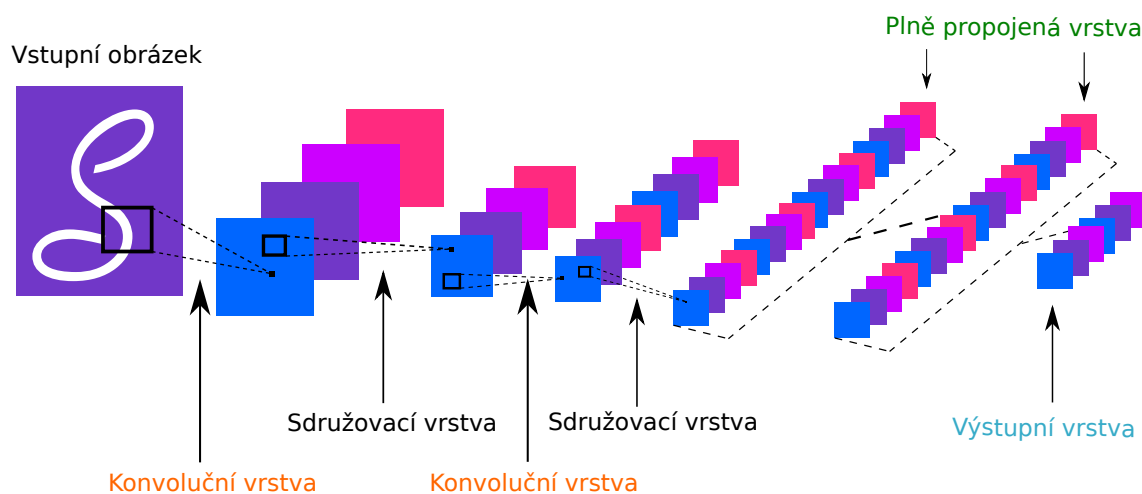
Jedná se o speciální dopředné vícevrstvé neuronové sítě, které zaujaly velký význam ve zpracovávání obrazu, jelikož díky svým vlastnostem dokázaly řešit velice efektivně úlohy zaměřené na obrazové zpracování. V současné době jsou široce uplatňované v hlubokém učení, zejména pro klasifikaci obrazu do více tříd, rozpoznávání apod.

Typickými znaky konvoluční neuronové sítě je velké množství neuronových vrstev, kde jsou neurony vzájemně propojeny vždy mezi jednotlivými vrstvami. Díky svým vlastnostem umožňují identifikovat objekt z obrazu na základě jeho podobnosti a to bez ohledu na posunutí, deformaci či úplně jinou velikost[14].

Konvoluční neuronové sítě mají svá určitá specifika. Mezi ty hlavní patří, že jsou hluboké, mají pevnou konvoluční strukturu propojení a disponují třemi typy vrstev:

1. Konvoluční vrstva
2. Sdružovací vrstva
3. Plně propojená vrstva

Architekturu konvoluční neuronové sítě lze vidět na obr. níže.



Obr. 1.5: Architektura konvoluční neuronové sítě.

### 1.4.1 Konvoluční vrstva

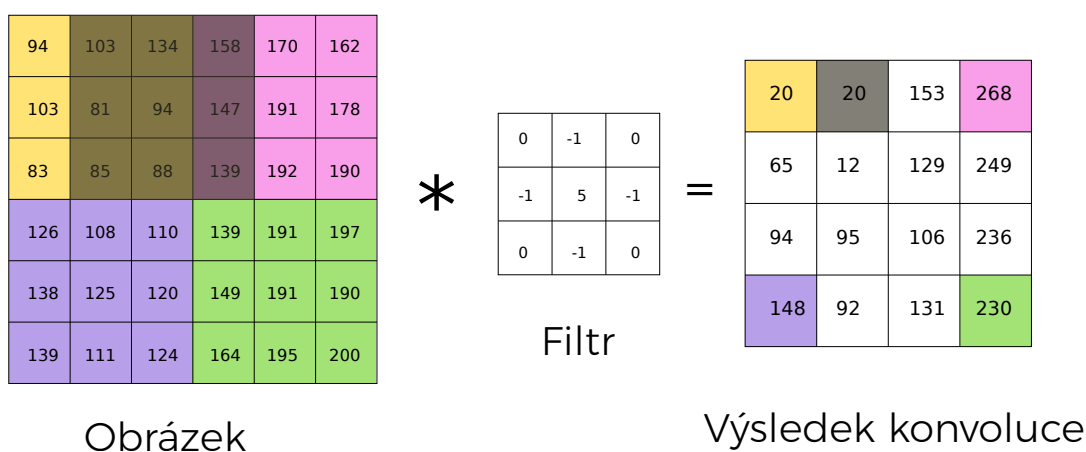
Aby byla neuronová síť schopna rozeznat jeden obrázek od druhého, postupuje velice podobným způsobem jako lidský mozek. Začne hledat znaky, které obrázek vystihují. Tyto vlastnosti resp. lokální charakteristiky obrazu má tedy každý obrázek jiné a samy o sobě zaručují rozeznání jednoho obrázku od druhého.

Vstupem konvoluční vrstvy je tedy **obrázek**, ze kterého jsou vytvořeny takové *mapy příznaků* - jedná se o výřezy obrazu např. o velikosti 3x3px, které jsou pak vstupem jednoho neuronu, který je díky sdílení vah dále rozkopírován pro každou podoblast. Z těchto map jsou poté vytvořeny filtry neboli konvoluční jádra viz. obr.1.6, které filtrují vstupní obraz pomocí konvoluce.

Konvoluce je diskretní matematická operace od níž vrstva dostala i své jméno. Tato operace se provádí nad celým obrázkem a díky ní je konvoluční vrstva schopna matematicky vypočítat míru shody obrázku s příznakem. Diskretní 2D konvoluci lze matematicky definovat vztahem 1.2[15]:

$$I(x, y) * h(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k I(x - i, y - j) \cdot h(i, j) \quad (1.2)$$

kde  $h(x, y)$  představuje *konvoluční jádro* a  $I(x, y)$  diskretní obraz. Jak již bylo nastíněno výše, *konvoluční jádra* jsou jakési filtry, pomocí kterých se filtruje obraz<sup>2</sup>. V konvoluční vrstvě je těchto filtrů více (podle počtu příznaků), protože každý má za cíl detekovat jinou část obrázku. Výsledkem poté je matice hodnot, neboli přefiltrovaný obrázek pomocí těchto jader.



Obr. 1.6: 2D konvoluce.

<sup>2</sup>Z důvodu překryvu byly na obr. 1.6 vyznačeny barevně jen některé oblasti konvoluce.

Při filtrování je také definován krok, kterým říkáme, o kolik se má filtrační maska posunout při konvoluci další části obrazu. Čím větší krok tedy, tím rychlejší filtrace, ale nižší přesnost.

Výstupních map příznaků, resp. obrázků přefiltrovaných těmito mapami, musí být stejný počet jako filtrů. Výstup konvoluční vrstvy poté slouží jako vstup sdružovací vrstvy.

Na obr. 1.6 je patrné, že při zpracování obrazu bylo použito velikosti kroku filtrace 1 - maska se posouvá vždy o jedno políčko a poté provádí výpočet znova.

### **Dilatovaná konvoluční vrstva**

Vedle klasické konvoluční vrstvy existují ještě tzv. dilatované konvoluční vrstvy. Hlavní rozdíl mezi nimi je ten, že dilatované ještě dále rozšiřují práci s konvolučními jádry. V knihovně `keras` (viz. kapitola 2.4.2) je možné z normální konvoluce udělat dilatovanou pomocí parametru `dilation_rate`. [16]

Dilatovaná vrstva dle nastavení rozděluje resp. vkládá mezeru mezi jednotlivé parametry konvolučního jádra. [17] Pokud by tedy bylo konvoluční jádro o velikosti 3x3 a dilatační parametr nastaven na 2, vznikne konvoluční jádro o velikosti 5x5 a to tak, že zůstane každý druhý řádek a sloupec jádra prázdný. Zde lze již vidět jasná výhoda dilatace. I s menším konvolučním jádrem lze pomocí dilatování dosáhnout větší zpracovávané oblasti se stejnými výpočetními požadavky, jako by se jednalo pouze o konvoluci s jádrem 3x3. Nemění se tedy ani celkový počet map příznaků.

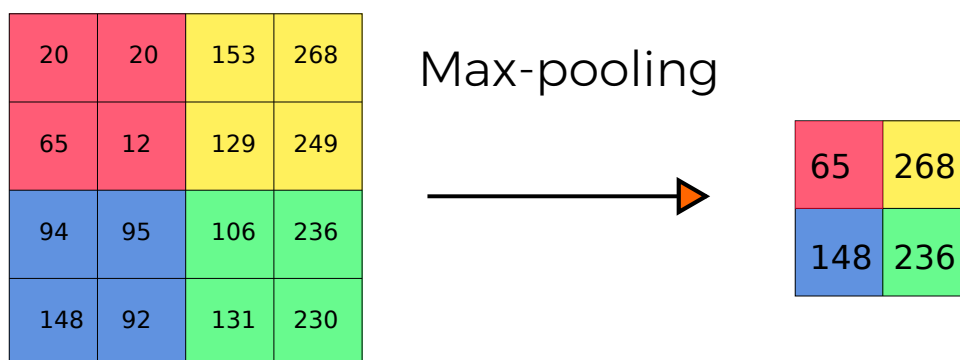
Ne vždy je tato vrstva ale přínosem, obzvlášť v situacích, kdy je nastaven vzrůstající dilatační faktor příliš agresivně a je třeba rozlišovat malé objekty [17]. Takto dochází ke generování nepřesných map příznaků, které velkou část důležitých znaků v obraze přeskakují.

### 1.4.2 Sdružovací vrstva

Sdružovací vrstva realizuje podvzorkování vstupních dat. Tato vlastnost má zásadní heuristický vliv na celou síť, protože výsledek vede k tomu, že se eliminuje velká část hodnot a zanechají se jen některé, díky čemuž se sníží počet operací pro vyšší vrstvy[18].

Nejpoužívanější funkcí pro tento účel se používá **max-pooling**, která podle velikosti okna vybere nejvyšší hodnotu v dané oblasti a tu zanechá, všechny ostatní jsou poté vyloučeny tj. agregace velikosti na jednu jedinou hodnotu. Obrovskou výhodou *max-poolingu* je, že se přímo podílí na zvýšení invariance vůči posuvu. Nejčastěji se používá velikost okna  $2 \cdot 2$  popřípadě  $3 \cdot 3$ , na obr.1.7 je při podvzorkování použita oblast o velikosti  $2 \cdot 2$ . Ve většině případů, typicky při segmentaci obrazu, dosahuje max-pooling nejlepších výstupů oproti ostatním sdružovacím funkcím.

V případech, kdy není nutné podvzorkovávat takto progresivně jako u max-poolingu, je zde alternativa **average-pooling**, který agreguje masku na jedinou hodnotu pomocí průměrování hodnot v okně. Tímto postupem dojde k tomu, že se do další vrstvy přenáší i část informace z okolních bodů, což může v konečném důsledku znamenat i „přesnější“ výstup.<sup>3</sup>



Obr. 1.7: Max pooling.

<sup>3</sup>V praxi záleží vždy na datasetu, tedy jaká data se budou klasifikovat a podle toho se odvíjí také použití.

### 1.4.3 Plně propojená vrstva

Výstupy z předchozích vrstev jsou v plně propojené vrstvě transformovány do jednorozměrného vektoru, který je poté vstupem každého neuronu výstupní vrstvy, kde se vypočítá konečný výstup. Každý neuron plně propojené vrstvy je spojen s všemi neurony z předcházející vrstvy. Plně propojená vrstva obsahuje stejný počet neuronů jako je počet klasifikačních tříd, do kterých se vstupní obrázek bude třídit.[19] V praxi vzniká takových plně propojených vrstev více - vytváří se síť neuronů, kde tato architektura má vstupní, výstupní a skryté vrstvy viz. část Princip fungování sítě.

Každá výstupní vrstva je ohodnocena mírou pravděpodobnosti výstupu, na jehož základě je poté rozhodnuto o výsledku. Součet pravděpodobností v plně propojené vrstvě je dána hodnotou 1.

#### Metoda dropout

Metoda slouží k předejití problému přetrénování sítě a využívá se u plně propojených vrstev.

Problém přetrénování spočívá v tom, že síť začne ztrácet svoje správně naučené obecné rysy a bude se více specializovat na daný problém, který bude sice umět řešit, ale s jeho obdobnými variantami si již nebude schopna poradit.

Princip metody *dropout* spočívá v deaktivaci části neuronů na vrstvě po každém trénovacím průchodu a jejich výsledným průměrováním po natrénování sítě.



## 1.5 Dávková normalizace

Dávková normalizace (*Batch-normalization*) je technika, kterou v roce 2015 představili Christian Szegedy a Sergey Ioffe z Google AI. Řeší problém tzv. kovariantního posuvu.

Při trénování neuronové sítě dochází neustále ke změně rozložení aktivací jednotlivých neuronů, což zpomaluje proces učení celé sítě, protože každá vrstva se musí přizpůsobit nové dávce dat dle kroku učení. Nejvíce se tento problém vyskytuje v mezilehlých vrstvách.

Problém kovariantního posuvu se do jisté míry dá řešit menším koeficientem učení a zvýšením počtu epoch. Díky dávkové normalizaci ale můžeme naopak potom koeficient učení zvýšit, celý proces učení tedy bude konvergovat rychleji s výslednou menší chybou.

V praxi se vrstva umísťuje mezi konvoluční či plně propojenou vrstvu a aktivační funkci.

$$\mu_{\mathcal{B}} = \frac{1}{m} \sum_{i=1}^m x_i \quad (1.3)$$

$$\sigma_{\mathcal{B}}^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad (1.4)$$

$$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad (1.5)$$

$$y_i = \gamma \hat{x}_i + \beta \quad (1.6)$$

Vztahem 1.3 je vypočítán průměr vstupní dávky a poté pomocí rovnice 1.4 stanoven rozptyl. Samotná normalizace dávky se vypočítá vztahem 1.5[20] díky stanovení statistik dávky v předchozích dvou krocích. Výsledek normalizace je dán rovnicí 1.6, kdy dochází ke škálování dávky a výslednému posuvu.[21]

Proměnné  $\gamma$  a  $\beta$  jsou nastaveny sítí během trénování spolu s původními parametry sítě.[22]

Normalizace udržuje střední aktivační hladinu kolem 0 a standardní odchylku blízko hodnoty 1.[23]

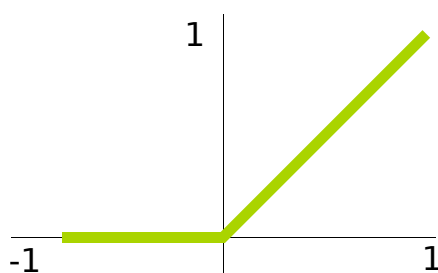
## 1.6 Aktivační funkce

Slouží k aktivaci neuronu na jeho výstup. Důležitým kritériem aktivačních funkcí je, aby produkovaly nelineární hodnoty. Důvodem je fakt, že pokud by lineárními byly, tak potom by celá neuronová síť byla lineární, čímž ztratíme veškerou výhodu neuronových sítí.

### 1.6.1 ReLU

ReLU (*Rectified Linear Unit*) je aktivační funkce definovaná vždy od 0, tudíž pokud narazí na hodnoty, které jsou záporné, automaticky je převede do svého rozsahu tj. na nulu. Zbylé hodnoty zůstávají netknuté. Konverguje rychleji nežli funkce *sigmoid* nebo *tanh* a není souměrná kolem nuly.[20] ReLU je definovaná vztahem 1.7:

$$f(x) = \max(0, x) \quad (1.7)$$

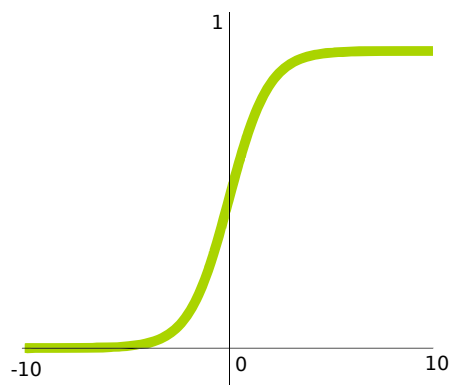


Obr. 1.8: ReLU

### 1.6.2 Sigmoid

Jedná se historicky o nejpoužívanější aktivační funkci. Je dána rozsahem  $< 0, 1 >$  a není souměrná kolem nuly. Je definována vztahem 1.8[24]:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1.8)$$

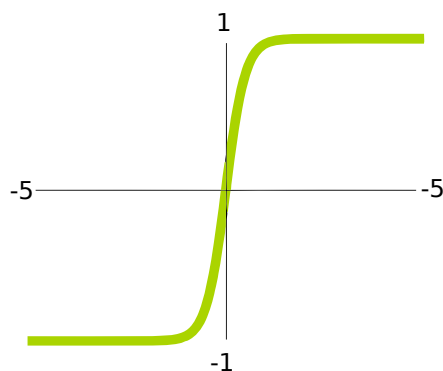


Obr. 1.9: Sigmoid

### 1.6.3 Tanh

Aktivační funkce s rozsahem  $\langle -1, 1 \rangle$ , která je souměrná kolem nuly. Dána vztahem [24]:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1.9)$$



Obr. 1.10: Tanh

### 1.6.4 Softmax

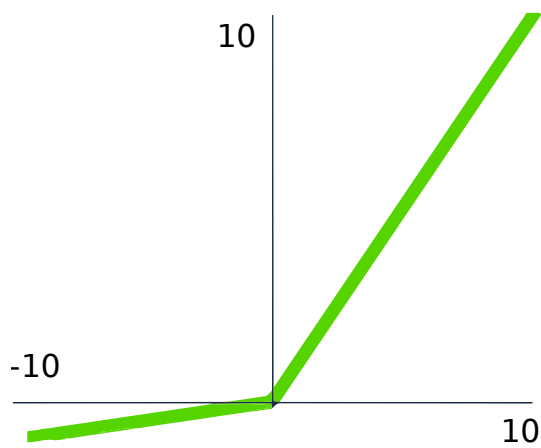
Používá se k vícetříde klasifikaci, kde je objekt přiřazen do jedné třídy. Suma všech pravděpodobností je rovna jedné. Všechny vypočítané pravděpodobnosti jsou vždy v rozsahu od nuly do jedné. Aktivační funkce *softmax* je dána vztahem [25]:

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (1.10)$$

### 1.6.5 LReLU - Leaky ReLU

LReLU vznikla modifikací původní funkce ReLU. Hlavním rozdílem oproti ní je ta, že LReLU nahrazuje nulovou část tj. interval  $[-\infty, 0]$  nízkým sklonem. Hlavním důvodem zavedení LReLU je ten, že nulový sklon v případě funkce ReLU může v určitých případech vést k pomalému učení např. když neuron využívá aktivační funkci `sigmoid` - některé z neuronů se nemusí vůbec aktivovat.[26] Funkce je dána vztahem[20]:

$$f(x) = \max(0.01x, x) \quad (1.11)$$



Obr. 1.11: LReLU

## 2 Analýza problému

?? V následující kapitole budou popsány současné metody segmentace obrazu, které používají tzv. hluboké učení. Dále jejich vlastnosti se zaměřením na použití pro segmentaci dat do více tříd. Součástí kapitoly bude také popis dat, které jsou přímo určené pro vícetřídní zpracování a princip jejich získávání.

### 2.1 Segmentace obrazu

V dnešní době je již nedílnou součástí zpracování obrazu právě segmentace, která je jednou z nejdůležitějších kroků při analýze obrazu. Hlavním cílem segmentace je proces dělení obrazu na části a rozlišení jednotlivých objektů v obraze, s kterými se poté dále pracuje, například členění jednotlivých objektů k určité třídě podobně vysegmentovaných objektů. Segmentace se při dalším zpracování hojně využívá ke klasifikaci objektů, vychází hlavně z globální znalosti obrazu např. barvy, tvaru, apod. V současné době existuje velké množství segmentačních metod. Velká část z nich je nastíněna v následující kapitole.[27]

Proces segmentace je dělení obrazu na podobrazy, které splňují určitá kritéria. Splňujícím kritériem může být buď jediné nebo i jejich množina. Mezi takové kritéria patří:

- Pixely v segmentované části resp. podobrazu mají stejnou úroveň šedi, barvy, jasu apod.
  - Úroveň šedi, barvy, jasu apod. daných pixelů v segmentované části resp. podobrazu se neliší více nežli o velikosti předem stanovené hodnoty.
  - Odchylka úrovně šedi jednotlivých pixelů v podobrazu je dostatečně malá apod.
- [15]

## 2.2 Současné metody segmentace obrazu

K segmentaci lze přistoupit celou řadou způsobů, kdy můžeme segmentační algoritmy rozdělit na metody zaměřující se na detekci hran v obraze, kde se využívá faktu, že existuje významný rozdíl okolních pixelů, což umožní hranovým detektorům identifikovat lokální hrany. Detektory hran jsou algoritmy, jejichž výstupem je množina hran v obraze.

Mimo metod založených na *detekci hran* existují také metody orientované na oblasti (regiony). Využívají detekování hran k tomu, aby mohly rozeznat oblasti v obraze. Mají ale jednu velkou nevýhodu, pokud by kontura byla v obraze porušena, tak k ohrazení celé oblasti dojít nemusí. [28]

Další metodou a zároveň jednou z nejstarších, ale také nejjednodušších je *metoda prahování*. Lze ji využívat samostatně nebo jako součást dokonalejších metod. Prahování při segmentaci využívá rozdílných úrovní intenzity objektu a pozadí. Na základě stanovení prahu se poté určuje příslušnost pixelu buď k pozadí nebo objektu.[29]

Pokud bychom využívali toho, že metody mají určitou znalost obrazu resp. segmentovaných objektů, potom se jedná o *znalostní metody*. Tyto metody mohou segmentaci velice ulehčit, protože využívají tzv. atlas modelů či předloh segmentovaných objektů, které jsou generovány na základě trénovacích dat automaticky. V případech, kdy to je možné, lze předlohy vkládat ručně díky bohaté lidské zkušenosti, což poté povede k přesnější segmentaci.[15]

Někdy segmentační techniky obsahují vlastnosti každé z již zmíněných a není možné je tedy jednoznačně zařadit. Tyto metody označujeme jako *hybridní* a patří sem metody založené na morfologii, amplitudové projekci nebo **neuronových sítích**. [29]

### 2.2.1 Detekce hran

Na segmentaci pomocí detekce hran je založena celá řada segmentačních algoritmů. I když tato metoda patří do nižší úrovně zpracování obrazu, má v segmentaci stále své místo. Hranou jsou označeny body, ve kterých se prudce mění jas, což v praxi znamená, že jde o hranice objektu, pozadí apod. Každá hrana je vedle prudkého výkyvu jasu určena také směrem a velikostí. Ideální hranou pro účely modelování by tedy byla skoková funkce, ale v reálných datech jsou změny jasu hrany téměř vždy postupné.

Pokud by se tedy na hranu pohlíželo jako na funkci, která pro daný pixel vrací její velikost intenzity, lze takovou hranu popsat průběhem funkce na obr. 2.1, kde lze vidět více typů hran[30]. Oranžová hrana značí typ hrany *roof*, fialová *line*, zelená *step edge* a červená *ramp*.



Obr. 2.1: Jednotlivé typy hran.

Jak lze vidět, hrana může nabývat různých podob. Důvodem je ten, že určení přesné polohy hrany je složitý proces, jelikož zde jsou faktory, které přímo určování hrany ovlivňují. Jedná se zejména o šum, kvůli kterému není snadné rozhodnout, zdali se změna intenzity vtahuje na hranu nebo právě na šum. Tento problém se da vyřešit předzpracováním pomocí filtru.[27]

Hlavní metody detekce hran využívají *první derivace* popřípadě *druhou derivaci*. V okrajových případech, kdy tyto metody použít nelze, např. tam, kde dochází k detekování hranic dvou sousedících textur, se používají *statistické metody* nebo *srovnání se vzorem*.

Při použití první derivace vycházejí metody z předpokladu, že k největší změně intenzity dochází v místech hrany, zatímco v ostatních oblastech předpokládají, že je výkyv intenzity nulový. Takové metody se také označují jako gradientní. První derivace se získává rozdílem okolních pixelů diskrétního obrazu. Postupuje se zejména tak, že se řádky a sloupce vypočítávají zvlášť (platí pro nejjednodušší případy). [15] V případě použití druhé derivace se využívá faktu, že je daleko jednodušší nalézt bod, kde dochází k průchodu nulou nežli hledání extrému. Využívá se zejména v případech, kdy není nutné znát velikost a směr hran.

## 2.2.2 Prahování

Metoda prahování (thresholding) je velice populární díky své jednoduchosti, může být používána jak samostatně, tak jako součást složitějších metod. Celá metody je založena na předpokladu, že objekty a pozadí mají rozdílný stupeň intenzity a díky tomuto předpokladu se stanovuje tzv. *práh*.

Jedním z nejtěžejnějších úkonů celé metody je tudíž stanovení adekvátního prahu  $T$ , na základě něhož se srovnávají vstupní pixely obrazu a zjišťuje se, zdali

splňují podmínku prahu, pokud ano je pixel označen jako pixel objektu, pokud ne jedná se o pozadí. Algoritmizace celé metody by mohla vypadat dle vztahu viz. 2.1,

$$if(g(i, j) \geq T) : \quad (2.1a)$$

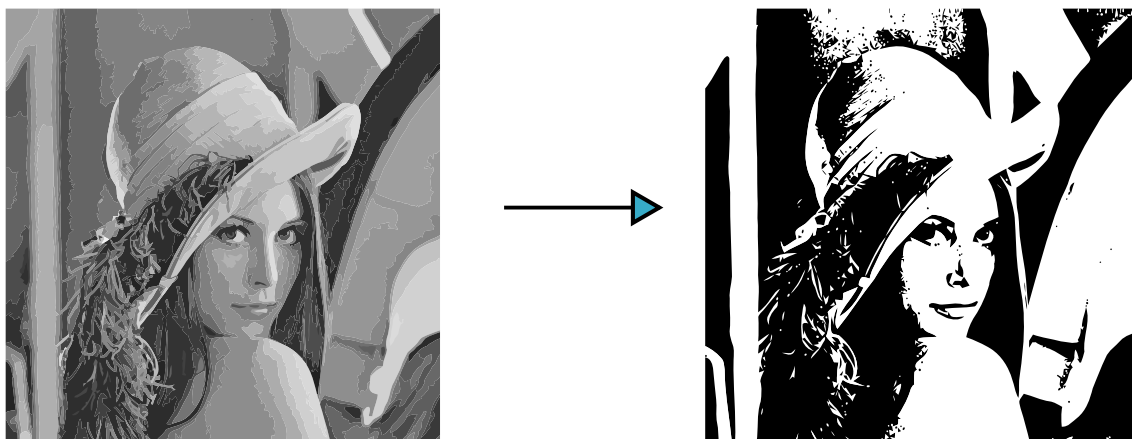
$$f(i, j) = 1 \quad (2.1b)$$

$$else : \quad (2.1c)$$

$$f(i, j) = 0 \quad (2.1d)$$

kde  $g(i, j)$  značí vstupní obraz a  $f(i, j)$  výstupní obraz s prahem  $T$ . Existuje vícero způsobů, jak práh  $T$  nastavit. Jedním z nich je *globální prahování*, kdy je jedna hodnota prahu nastavena pro celý obraz. U tohoto postupu je velká nevýhoda v případě velkých výkyvů intenzit v obrázku (např. různé části obrázku budou jinak osvětleny), která se ale dá vyřešit předzpracováním obrazu určitým filtrem.[27].

Výsledek prahování lze vidět na obrázku vpravo při použití prahu stupně šedi 127 na škále 0-255.<sup>1</sup>



Obr. 2.2: Prahování.

V případech, kdy globální prahování je zcela nevhodné, je možné použít metodu *adaptivního prahování*, které se umí daleko lépe vyrovnat s velkou různorodostí obrázku. Obraz je rozdělen na menší části a pro každou takovou část je určen práh. Vypočítává se z hodnot jasu pomocí mediánu případně průměrováním nebo z průměru maximálních a minimálních hodnot. Dále je možné na výpočet prahu použít iterativní metodu, ve které se využívá konvergence prahových hodnot.[31].

---

<sup>1</sup>Na původním bitmapovém obrázku „Lena“ bylo nejprve provedeno prahování a poté byly oba bitmapové obrázky vektorizovány.



### 2.2.3 Metody založené na regionech

Metody založené na regionech namísto hran detekují oblasti o homogenních vlastnostech. Přináší to velkou výhodu hlavně v případech, kdy je v obraze hodně šumu. Regionální metody se s tímto problémem dokáží vypořádat lépe než metody založené na detekci hran. Homogenní vlastnosti mohou být např. barva, tvar, úroveň šedi apod.[15] Stejně jako předešlé segmentační techniky i metody založené na regionech disponují více způsoby jak oblasti detekovat.

#### Metoda spojování oblastí

Nejdříve dochází k rozdělení obrazu na podoblasti, pro která se pak definuje kritérium pro jejich spojování resp. spojení jednotlivých pixelů. Kritériem jsou zde myšleny společné znaky náležící určitému objektu v obraze. K dokončení operace spojování dochází ve chvíli, kdy již není možné spojit žádnou oblast. Sdružené podoblasti pak reprezentují hledaný podobraz.[32]

#### Metoda štěpení oblastí

Metoda štěpení postupuje opačným způsobem jako metoda spojování oblastí. Na začátku je jedna velká oblast, která se pak dále dělí na co největší počet navzájem nezávislých podoblastí.

#### Metoda watershed transform

Neboli metoda rozvodí. Metodu lze také zařadit do hybridních metod díky svým vlastnostem. Princip metody má geografický základ, kdy je obraz chápán jako reliéf, který je pak zaplavován vodou. Rozvodí jsou u této metody chápány jako hranice mezi regiony.

Metoda postupuje tak, že nejprve nalezne lokální minima a poté začne povodí zaplňovat vodou. Jak postupně hladina sloupá, tak by mohlo dojít k tomu, že by se dvě povodí slily. V takovém případě se mezi nimi vytvoří hráz. Metoda skončí v momentě dosažení maxima daného terénu.[29][33]

#### Metoda narůstání oblastí

Nejjednodušším způsobem jak vytvořit segmentovanou oblast je metoda narůstání oblastí (*region growing*). Jak již název napovídá, jedná se o seskupování oblastí sousedních pixelů s podobnou amplitudou.

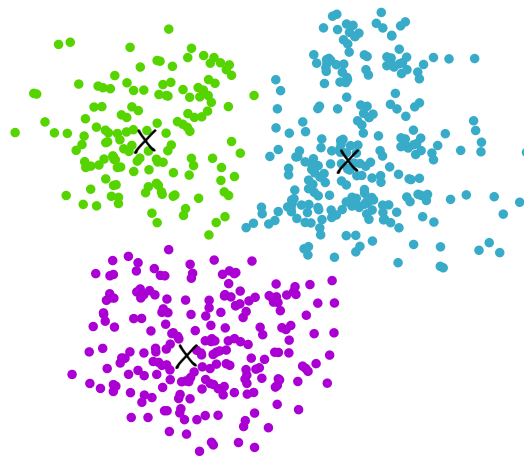
Při vytváření počátečního regionu se zvolí náhodný pixel tzv. semínko, které je považováno za počáteční bod regionu. Postupným připojováním sousedních pixelů dochází k rozrůstání oblasti a to až do doby, dokud není splněno určité kritérium

homogeneity.[27] Těchto semínek je zvoleno více a cílem je z nich vytvořit souvislé oblasti, které podléhají zvoleným kritériím. Rozrůstání semínka může probíhat buď vždy přes hranu tj. sousední pixely, tudíž rozrůstání oblasti probíhá ve čtyřech směrech nebo je rozrůstání ještě obohaceno o diagonální sousedy, potom se tedy semínko rozrůstá do osmi směrů. Další krok rozrůstání se děje až v momentě, kdy dojde k výpočtu rozdílu mezi počátečním bodem semínka a jeho sousedem, pokud pak je tato vypočtená hodnota menší než je kritérium, dojde ke sloučení pixelů. Jako nové semínko je zvoleno to s nejmenším rozdílem. Poté dojde k výpočtu nové hodnoty počátečního bodu jako průměr hodnot regionu.[32]

## Shluková analýza

Jedná se o obecnou statistickou klasifikační metodu, která při zpracování obrazu sdružuje pixely podobných vlastností do tzv. shluků. Každý shluk má poté nějaké společné vlastnosti, které jsou vždy odlišné od vlastností u jiných shluků. Shlukování je použito například u nerunových sítí v případě učení bez učitele. [33]

Jedním ze způsobů, jak data shlukovat, je metoda *k-means*. Jedná se o iterativní algoritmus, kde počet shluků  $k$  je předem dán. Jsou zavedeny centrální body (středů shluků), které reprezentují shluk a jsou také ve stejném prostoru jako vstupní množina dat. Centrální body jsou obvykle voleny náhodně nebo s využitím nějaké heuristiky. Kolem tohoto centrálního bodu se pak ostatní body zařazují dle nejmenší vzdálenosti. Jakmile dojde ke klasifikaci dle stanoveného kritéria, tak je poté i přepočítán středový bod.



Obr. 2.3: K-means.

## 2.2.4 Znalostní metody

Pokud metody využívají již dříve získané znalosti objektů v obraze, potom se řadí do znalostních metod, jelikož ostatní metody tyto informace k dispozici nemají.

Získané znalosti jsou interpretovány jako modely nebo šablony objektů. Princip znalostních metod poté spočívá v tom, že tyto šablony/modely objektů srovnávají s přiloženým obrazem a hledá se shoda.[15] Srovnávat lze také regiony nebo hledání shody pouze části obrazu s částmi určitého vzoru. Vlivem šumu, zkreslení apod. ale nelze očekávat absolutní shodu, úspěchem resp. shodou v tomto případě bude vhodné maximální kritérium.[33]

### Active Appearance Models

Metoda se využívá při ztvárnění lékařských snímků, jejich detekci či rozpoznávání obličejů apod. Princip spočívá v tom, že se nejprve z vysegmentovaných trénovacích dat vytvoří statistický model, který je pak porovnáván s objektem v obraze.[33] Model se skládá z kontury a textury, které jsou uvnitř tvaru. Výsledkem je pak zjištění transformace modelu na vstupní obraz, která je řešena regresní analýzou, díky které dochází k minimalizaci rozdílu mezi vstupním obrazem a tím vytvořeným ze statistického modelu. Velkou výhodou metody je rychlé porovnávání resp. prohledávání obrazu. [34]

## 2.2.5 Hybridní metody

Jedná se o metody, které nelze klasifikovat do již zmíněných kategorií výše, jelikož využívají některé prvky předchozích typů metod tj. kombinaci přístupů.

### Metody založené na neuronových sítích

Metody popsané výše vždy postupovaly dle určitého předem daného algoritmu, řešící určitý typ problému určitým způsobem. V případě neuronových sítí je zcela vynecháno zaměření se na konkrétní úlohu, jedná se o obecný návrh, kterému stačí předložit vstupní hodnoty a jemu odpovídající výstupy. Algoritmizace je zde nahrazena strojovým učením. Síť se učí daný problém díky konfrontaci se správnými výstupy, čímž získávají markantní výhodu oproti jiným metodám.

Jak již bylo nastíněno v předchozí kapitole, neuronové sítě jsou velice flexibilní nástroj a v případě změny problému, na který jsou aplikovány, jsou schopny rychle reagovat díky možnosti učení. Smyslem neuronových sítí je přiblížit zpracování obrazu tak, jak jej zpracovává lidský mozek.

Konvoluční neuronové sítě postupují při segmentaci obrazu tak, že si nejprve ze vstupního obrázku sestaví sadu příznaků, kterými pak filtruje vstupní obraz pomocí

konvoluce, která nám udává míru shody. Výstupem vrstvy je potom stejný počet vrstev jako příznaků, protože se obrázek filtroval všemi příznaky a tedy disponuje více mezivýsledky, které slouží jako vstup následující sdružovací vrstvy.

Sdružovací vrstva tyto konvoluční vrstvy zmenší, přičemž se snaží zanechat maximum přínosných informací. Díky této vrstvě dojde k zásadnímu zmenšení vstupních dat a snížení paměťových nároků.

Poslední vrstvou je plně propojená vrstva, která řídí vyhodnocení zpracovaných dat. Tvoří logiku konvolučních neuronových sítí. Rozhoduje především o klasifikaci objektů do skupiny případně i skupiny, do kterých je objekt zařazen.

Po klasifikaci dochází již k sestavení obrazu a výsledné segmentaci.

**LeNet-5** Jedná se o sedmistupňovou konvoluční neuronovou síť, kterou navrhl již v roce 1998 Yann LeCun a jeho tým. Architektura vyniká svojí jednoduchostí, což byl také jeden z hlavních důvodů jejího dalšího širokého použití.

Síť se skládá ze dvojice konvolučních a poolingových vrstev, kterou jsou následovány podvzorkovací vrstvou, dále dvojicí plně propojených vrstev a nakonec klasifikátorem softmax.[35]

Její první použití sloužily k rozpoznání číslic, které byly digitalizovány do formátu 32x32px ve stupni šedi. S vyšším rozlišením vstupních obrázků má síť velké paměťové nároky.

**AlexNet** Síť má podobnou architekturu jako LeNet, ale oproti ní, jde více do hloubky, kde s každým zanořením se vyskytuje také vyšší počet filtrů. Po každé plně propojené nebo konvoluční vrstvě následuje aktivace ReLU

AlexNet se skládá z pěti plně propojených a pěti konvolučních vrstev. První dvě konvoluční vrstvy jsou následovány těmi sdružovacími. Následně třetí, čtvrtá a pátá vrstva jsou spojeny přímo. Po poslední páté vrstvě následuje sdružovací vrstva. Výstup sítě je realizován pomocí dvojice prvně propojených vrstev a klasifikátorem softmax. Po každé konvoluční a plně propojené vrstvě je aplikována funkce ReLU.[36]

**GoogleNet/Inception** Sít se skládá z celkem 27 vrstev z toho 5 sdružovacích (poolingových).[37] Celá sít se významně liší např. od AlexNetu nebo VGGNet architektury. Střed sítě zahrnuje konvoluci 1x1 a namísto plně propojených vrstev na konci sítě (na výstupu) je použita globální sdružovací vrstva. [38] GoogleNet také implementuje nový prvek, čímž se významně odlišuje od předšlých architektur a to je tzv. **inception module**, který využívá dávkovou normalizaci a je založený na několika malých konvolucích, díky čemuž je možné významně snížit počet parametrů.[37]

**VGGNet** Neuronová sít je složená z 16 konvolučních vrstev(v hlubší specifikaci může být tento počet i vyšší). Využívá pouze konvoluci 3x3 ale velký počet filtrů. Konvoluční krok je nastaven fixně na 1px. Stejně tak prostorové zarovnání pro konvoluce je 1px. Sdružování tj. **max-pooling** je prováděno přes okno o velikosti 2x2px s krokem 2.

Sadu konvolučních vrstev následuje trojice plně propojených vrstev s rozdílnými počty kanálů. Jejich nastavení je v síti VGGNet pro všechny konfigurace stejná.

Poslední vrstva obsahuje funkci softmax a všechny skryté vrstvy neuronové sítě používají jako aktivační funkci ReLU.[39]

**ResNet** Základy sítě ResNet jsou inspirovány sítěmi VGGNet, oproti nim ale disponují vyšším počtem konvolučních vrstev tj. 34. Oproti VGGNet má menší počet filtrů.

Konvoluční vrstvy mají filtry o velikosti 3x3. V síti musí být splněn fakt, že velikost výstupní mapy příznaků musí odpovídat stejnému počtu filtrů. Pokud v síti dochází k dělení map příznaků, potom se počet filtrů násobí. Navíc jsou mezi skupinky konvolučních vrstev přidány dopředná spojení. Díky takovým přidaným spojeníům je možné trénovat neuronovou sít i s větším počtem konvolučních vrstev aniž by architektura sítě byla složitější např. než sít VGGNet.

Podvzorkování je realizováno pomocí konvolučních vrstev s krokem 2. Plně propojená výsledná vrstva sítě má aktivační funkci softmax.[40]

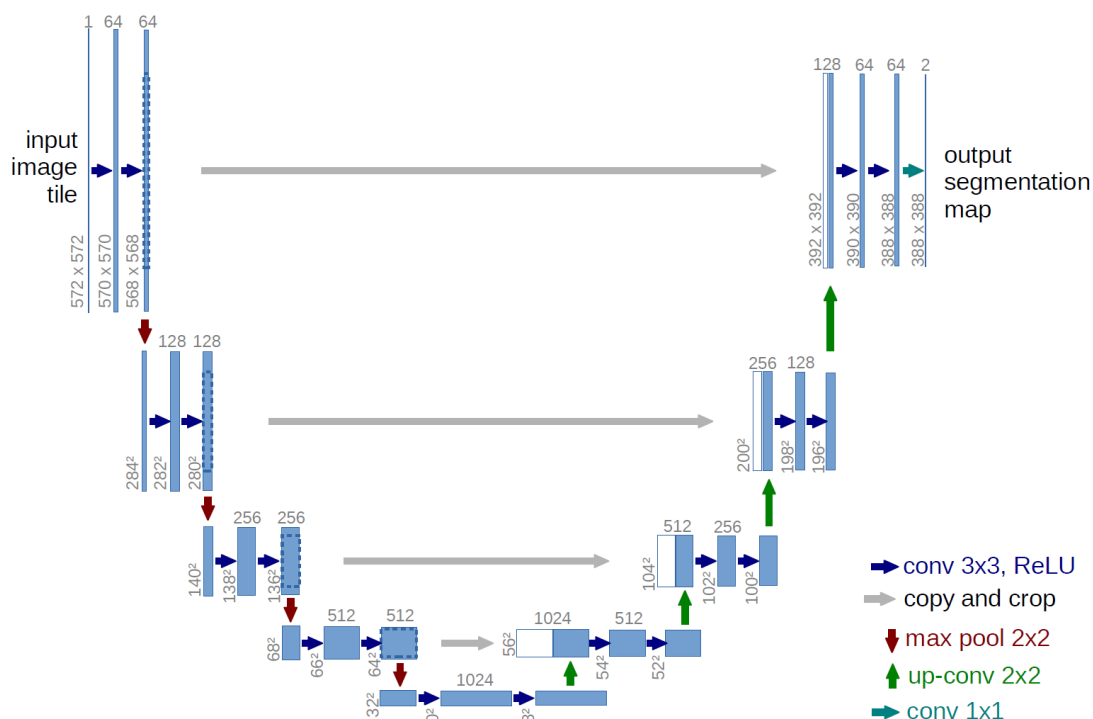
## 2.3 U-net

Jedná se o konkrétní podobu FCN (Plně konvoluční síť - *Fully Convolutional Network*) resp. její modifikaci, která provádí segmentaci obrazu. U-net architektura přináší vylepšení především pro lepší zpracování biomedicínských obrazů, jelikož dokáže lépe manipulovat s větším rozlišením vstupních obrázků s efektivnější paměťovou náročností.

Architektura byla modifikována a rozšířena tak, aby byla schopna i s menší množinou trénovacích dat dosahovat přesné segmentace. Jednou z nejdůležitějších modifikací také je, že v nadvzorkovací části je obsaženo velké množství map příznaků, které umožňují propagovat kontextové informace do vyšších vrstev. Mezi další charakteristické vlastnosti patří, že využívá symetrické rozšiřovací cesty, díky kterým umožňuje přesně lokalizovat jednotlivé pixely.

### 2.3.1 Architektura

Architektura UNET se skládá ze dvojice cest neboli stran, kde v levé části dochází k zanořování a v pravé postupné skládání ze zanořených vrstev. Zanořování se skládá z opakované aplikace konvoluce s aktivací ReLU a dále ze zmenšení neboli podvzorkování dat (poolingu). Naopak vynořovací strana poté aplikuje opačný proces, tedy nadvzorkování vzorků.



Obr. 2.4: Architektura U-net. Převzato z [2]

V každém podvzorkovacím kroku se zdvojnásobuje počet příznaků. Celkem má síť 23 konvolučních vrstev, které jsou rozděleny do pěti úrovní zpracování jak pro podvzorkování (pomyslné schody dolů), tak pro nadvzorkování (pomyslné schody nahoru). Po načtení vstupního obrázku následuje dvojice konvolučních vrstev  $3 \times 3$  a jedna podvzorkovací vrstva  $2 \times 2$ . Jednotlivé úrovně, tak jak postupně dochází k zanořování, jsou spojené s výstupními úrovněmi. Toto spojení slouží k překopírování a ořezání dat, což je pak vhodné při rekonstrukci výsledného segmentovaného obrazu. S každým přechodem do nižší úrovně resp. zanoření dochází ke zdvojnásobení map příznaků díky sdružovací vrstvě tj. 64 se zdvojnásobí na 128 atp.[41] Účelem podvzorkovacích vrstev napříč všemi úrovněmi zpracování je ten, aby během *poolingu* získaly co nejvíce kontextových informací o vstupním obraze, které se pak využijí k provedení segmentace. Analogickou resp. opačnou operaci nadvzorkování lze vidět poté opět u vynořování, kdy se počet map zase dvakrát zmenšuje. Mezi částí, kde dochází k zanoření a vynoření, se nachází pomyslný most, který je tvořen dvěma konvolučními vrstvami a dochází zde k metodě *dropout* viz výše a generování vektoru. Most zde má roli plně propojené vrstvy[2]

Díky svým výše popsaným vlastnostem umožňuje U-net architektura využívat jako vstup obrázky o rozdílných velikostech. Důvodem je ten, že jediné parametry, které mají vliv na učení jsou konvoluční jádra (mapy příznaků) a jsou zcela nezávislé na velikosti vstupních obrázků. Jejich menší velikost je méně paměťové náročná, tedy i zpracování bude rychlejší. V případě větších obrázků tomu bude sice naopak, ale zase se bude dosahovat přesnějších výsledků.[41]

## 2.4 Vývojové prostředí

K tomu aby bylo možné pracovat s architekturou U-net, je nutné mít připravené běhové prostředí. Jedná se o knihovny, díky kterým se dají vstupní data zpracovávat a poté s nimi dále pracovat.

### 2.4.1 Tensorflow

Jedná se o software s otevřeným zdrojovým kódem, který je určen pro vysoce náročné numerické operace. Díky své flexibilní architektuře umožňuje výpočty napříč různými platformami od obyčejných procesorů přes výkonné grafické karty až po speciální hardwarové výpočetní jednotky specializované na neuronové sítě. Díky API (Rozhraní pro programování aplikací - *Application Programming Interface*) lze provádět výpočty nejen na běžných počítačích, ale i clusterech, serverech a mobilních telefonech. API také poskytuje základní stavební bloky pro vytvoření trénovacích modelů.

Tensorflow byl původně vyvíjen výzkumným týmem a inženýry z firmy Google, která se zabývala strojovým učením a neuronovými sítěmi. Software je užitečný ale i v dalších vědních oblastech.[42]

### 2.4.2 Keras

Knihovna Keras je stejně jako Tensorflow software s otevřeným zdrojovým kódem. Jeho autorem je François Chollet z firmy Google. Je navržen tak, aby běžel pomocí některé z knihoven Tensorflow, Microsoft Cognitive Toolkit (CNTK) nebo Theano.

Keras je primárně určen a navržen tak, aby umožňoval rychlé experimentování s hlubokými neuronovými sítěmi. Velký důraz klade na to, aby byl rozšiřitelný, modulární a uživatelsky přívětivý. Podporuje jak konvoluční neuronové sítě, tak rekurentní sítě. Díky vhodně navrženému API, které následuje doporučené postupy tj. široce používané návyky při jeho návrhu, je významně zredukována kognitivní zátěž. [23]

### 2.4.3 Numpy

Knihovna NumPy se specializuje na zpracování numerických dat pomocí multidimenzionálních polí označované jako `ndarrays`. Je napsaná v pythonu a umožňuje rozšířit pole o operace nad jednotlivými prvky v poli, které python sám o sobě nenabízí. Například pokud je nutné použít nějakou lineární operaci nad polem pro jeho další použití, tak není nutné pole nejdříve předzpracovávat.



Pole mohou být také modifikovány dynamicky v případě, kdy je nutné se zbavit z určitého pole nějakých hodnot. Není nutné vytvářet nové pole, ale stačí použít masku na pole stávající.[43]

#### **2.4.4 Scikit**

Scikit neboli scikit-image je opět otevřená knihovna se zdrojovými kódy, která je určena ke zpracování obrázků v programovacím jazyce python. Autorem je Stéfan van der Walt.

Scikit-image umožňuje ukládání, načítání a úpravu obrázků. Dále obsahuje také algoritmy transformace, detekce příznaků, segmentace obrazu (např. pomocí prahování), analýzy, filtrování (zostření obrazu, odstranění šumu apod.) apod. Knihovna je navržena tak, aby byla kompatibilní i s dalšími knihovnami jako je např. SciPy nebo NumPy. [44][43]

## 2.5 Vstupní data

Následující část pojednává o přístroji, kterým jsou data snímány. Dále o typech dat, jejich formátu a způsobu, jakým se budou provádět veškeré operace nad takovými daty. Dále také způsob řazení dat do více tříd, proces zjednodušení a informace k manuální segmentaci tj. referenčním maskám.

### 2.5.1 Magnetická rezonance - MRI

Magnetická rezonance se v současné době řadí mezi jedny z nejmodernějších vyšetřovacích metod. Slouží pro přesné zobrazení částí lidského těla bez použití rentgenových paprsků, které jsou nahrazeny silným magnetickým polem. Principiálně magnetická rezonance dokáže interagovat s atomy vodíků, který je v lidském těle zastoupen v obrovském množství díky vodě. Silné stacionární magnetické pole interaguje s vysokofrekvenčními radiovými vlnami a dochází tak k detekování indukovaných proudů, které se poté převádí na obraz. Rozdílné intenzity znamenají různé hustoty atomů vodíku. Snímání částí lidského těla probíhá po řezech (tenkých vrstvách), které mají obvykle tloušťku do 5mm. Velkou výhodou MRI je také to, že kosti nijak nenarušují obraz. Tato vlastnost plyne z podstaty měření, jelikož magnetická rezonance se využívá především u měkkých tkání apod.

Doposud nebyly prokázány škodlivé účinky magnetické rezonance. Využívá se především k vyšetření mozku, míchy, dále při neurologických či onkologických onemocněních.

### 2.5.2 Popis zpracovávaných dat

Zvolený dataset, nad kterým se bude v následující části práce pracovat, je složen z více typů skenů magnetické rezonance. Data mají velikost voxelu  $0.958\text{mm} \times 0.958\text{mm} \times 3.0\text{mm}$  a všechny jsou zarovnané. Pole zkreslení bylo opraveno pomocí algoritmu *N4ITK*. Dataset je součástí výzvy *MRBrainS18*.

V některých snímcích T1-IR se ve spodní části objevují oblasti nebo objekty (šum), který se objevuje často právě při skenování a může mít vliv na přesnost, proto s ním musí automatická segmentace počítat.

Formát dodávaných dat je typu NIFTI (IT iniciativa pro neurozobrazování - *Neuroimaging Informatics Technology Initiative*) s koncovkou *.nii*. Důvodem zavedení takového lékařského formátu byl ten, že jiné typy dat nedisponovaly informacemi o orientaci v prostoru, tudíž data nemohla být jednoznačně interpretována.

Trénovací data se skládají ze 7 subjektů, ke kterým je přiložena i manuálně vytvořená reference s dělením do 11 tříd.[4] Třídy jsou definovány v tabulce 2.1

Štítek	Popis
0	pozadí
1	kortikální šedá hmota
2	bazální ganglia
3	bílá hmota
4	léze bílé hmoty
5	mozkomíšní mok (cerebrospinální tekutina CSF)
6	mozkové komory
7	mozeček
8	mozkový kmen
9	infarkt
10	ostatní

Tab. 2.1: Úplná tabulka segmentovaných tříd.[4]

Ovšem cílem výzvy je automatická segmentace do tříd 1 - 8, zbylé budou z vyhodnocení vyloučeny tj. nebudou nijak ovlivňovat výslednou přesnost a lze je označit libovolným štítkem. Štítek (třída) 0 neboli pozadí nebude považován za samostatný objekt.

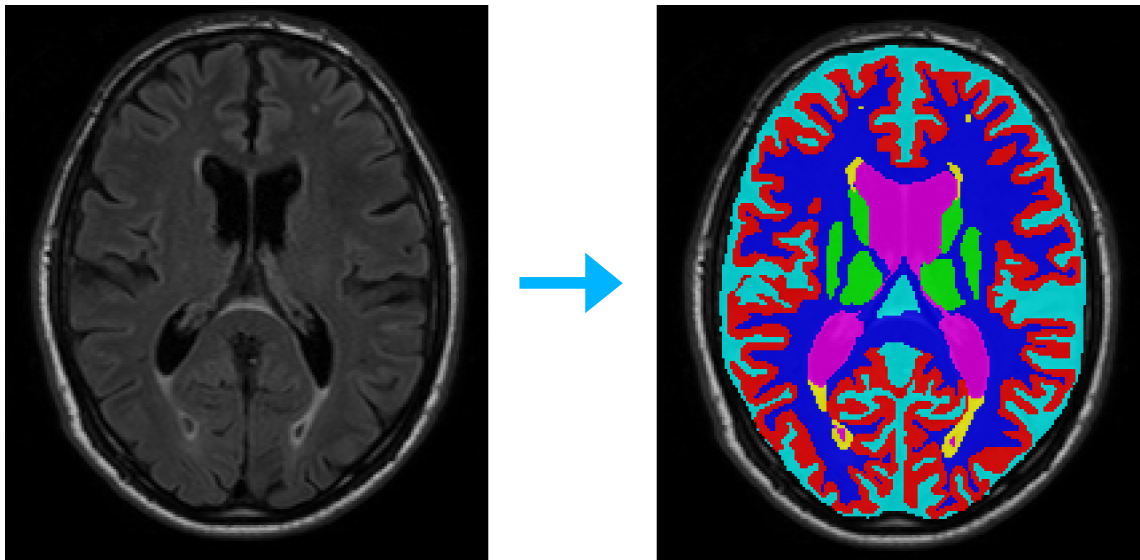
Algoritmy, které segmentují pouze šedou hmotu, cerebrospinální tekutinu a bílou hmotu by měly sloučit třídy následovně: 1 a 2, 3 a 4, 5 a 6 tj. celkem tedy 1-6. Výstup bude označeno jako 0, šedá hmota jako 1, bílá hmota pomocí štítku 2 a cerebrospinální tekutinu jako 3. Mozeček a mozkový kmen (7 a 8) budou z hodnocení vyloučeny a budou zařazeny zvlášť.[4]

### Poznámky k manuální segmentaci

- Na skenování léze (poranění) bílé hmoty byl použit sken FLAIR
- Vnější okraj CSF (Cerebrospinální tekutina resp. mozkomíšní mok - *Cerebrospinal fluid*) nebo *mozkomíšní mok* byl segmentován metodami:
  - T1(vážený)
  - T1(vážený)-IR (inversion recovery)
- Všechny ostatní struktury byly segmentovány pomocí T1(vážený) o velikosti voxelu  $0.958\text{mm} \times 0.958\text{mm} \times 3.0\text{mm}$
- V CSF je také zahrnut cerebrální falx.
- Segmentace CSF zahrnuje také vrchní sagitální sinus a transversální siny.[4]

## Vizualizace segmentace

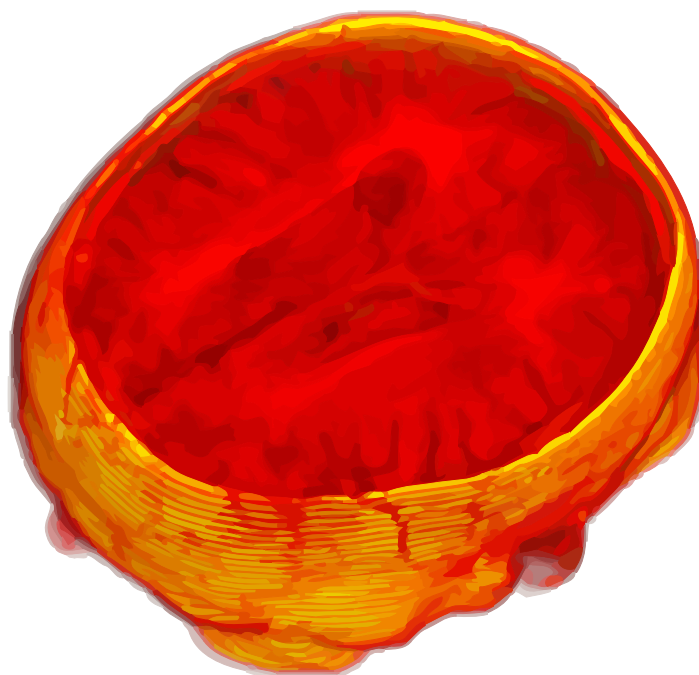
Vizualizace vícetříde segmentace lze vidět na obr. 2.5. Jak již bylo nastíněno výše, různé části mozku se označují jiným štítkem. Na řezu vpravo lze vidět tyto štítky obarvené pro lepší orientaci v segmentaci. Obrázek napravo je zároveň i ukázkou referenční manuální segmentace, podle které se bude celá konvoluční neuronová síť učit a upravovat svoje výstupy.



Obr. 2.5: 2D obrázek řezu mozku.

## Vizualizace 3D skenu

Jednotlivé snímky lze rekonstrukci poskládat tak, jak byl veden sken. Lze pozorovat jednotlivé řezy lebkou (žluté okraje) a průřez mozkem (rudá část). Obrázek 2.6 je realizován skenem FLAIR, výsledek skenu poté prošel i korekcí, kdy bylo opraveno pole zkreslení pomocí algoritmu viz výše. <sup>2</sup>.



Obr. 2.6: 3D obrázek řezů mozku.

---

<sup>2</sup>Renderování řezů do 3D bylo realizováno pomocí programu Aliza. Bitmapový obrázek byl poté vektorizován.

## 3 Implementace vícetřídé segmentace

Vícetřídou segmentaci lze řešit nepřehledným množstvím způsobů a také z toho důvodu bude následující kapitola pojednávat o více možnostech a přístupech, jak problém vícetřídé segmentace řešit. Tyto metody budou trénovat neuronovou síť nad stejně objemným datasetem viz. 2.5.2.

Postupy, které byly zvoleny pro řešení vícetřídé segmentace, se liší zejména ve způsobu, jakým bude na data pohlíženo a jakým způsobem budou zpracovávány, od čehož se také bude odvíjet následné řešení celé segmentace. Změny přístupu ve zpracování dat se poté musí též reflektovat i v samotném modelu konvoluční neuronové sítě.

První možností, jak vícetřídou klasifikaci provést, je že se vstupní velikost modelu bude shodovat s velikostí obrázku, tedy nebude nutné obrázky upravovat, co se do jejich velikosti týče. Počet vstupních dat bude odpovídat počtu 3D skenů. Model bude nastaven tak, aby byl schopen rozlišovat 10 různých tříd. Výhodou takového řešení je, že příprava dat je jednodušší.

Druhý přístup bude poněkud odlišný. K datům bude přistupovat tak, že si 3D skeny zmenší na menší části, které se pak mohou a nemusí překrývat. Jeden sken tedy nebude zpracován sítí během jednoho kroku, jako je tomu u řešení výše. Jeli-kož se data takto zmenší resp. jejich dimenze, tak vznikne velké množství fragmentů z původního velkého skenu, což je velice přínosné např. v případech, kdy je dat nedostatek. Počet kroků pro zpracování jedné hlavy bude tedy roven počtu 3D fragmentů vytvořených z původního nezmenšeného 3D skenu. Díky fragmentování 3D skenu lze použít při zpracování i překryv tj. posuvné okno, které se postará a rozložení hlavy na menší části. Překryvem jednotlivých kostek pak objem fragmentů naroste poměrově o velikost překryvu, což má poté též pozitivní vliv na celou síť.

### 3.1 Transformace vstupních dat

Vstupní data, jak již bylo nastíněno v předešlé kapitole, jsou dodávány ve formátu *.nii*. Pro účely zpracování byly data převedeny pomocí skriptu `dataset.py` do formátu *.png*, se kterým se bude dále pracovat. Velikost každého řezu je 240 x 240 pixelů. Každý sken se skládá ze 48 takových řezů. Dataset disponuje 7 různými 3D skeny, které jsou nasnímány v různých režimech. Výhodou formátu *.nii* oproti *.png* je ten, že obsahuje dodatečné informace o umístění řezu v prostoru, což zlepšuje jejich interpretaci zejména v případech, kdy je důležité se v nich rychle zorientovat.

Důvodem zvolení exportu do `.png` souborů ale byla jejich výhoda v dalším zpracování, jako je možnost prohlížení přímo po jednotlivých řezech nebo garance, že obrázky v sobě nebudou mít dodatečné informace, které se přímo dat jako takových netýkají a mohly by poté mít i negativní vliv na výsledek trénování.

### 3.1.1 Zpracování vstupních dat a proces výpočtů

Aby se dalo s jednotlivými obrázky neboli řezy mozku pracovat, bylo třeba tyto obrázky nahrát do numpy matic, které vedle uchovávání dat také disponují celou řadou operací nad těmito daty (viz. v kapitole 2.4.3), které přináší obrovskou výhodu při dalším zpracování.

### 3.1.2 Základní zpracování dat a načtení do numpy

Jak již bylo zmíněno výše, každý vstupní obrázek se skládá ze 48 řezů o velikosti 240 sloupců a řádků - zde je ale třeba ještě uvažovat kanál, který z důvodu jednoduchosti je pouze jediný. Paleta RGB je zde převedena na stupeň šedi<sup>1</sup>, čímž se nám počet kanálů zredukuje z 3 na 1. Proces zpracování probíhal tak, že se řez po řezu nahrál do matic a výsledná numpy matice měla rozměr pro jeden obrázek při úvaze jednoho kanálu ve formátu n-tice:

(počet\_skenů, pocet\_řezů, šířka\_řezu, výška\_řezu, kanál) což odpovídá tvaru: 48x240x240x1 resp. 7x48x240x240x1 pro 7 mozků.

U masek je situace poněkud složitější, protože je zde ještě nutné uvažovat vedle základních 4 rozměrů také třídy, podle kterých mají být trénovací data segmentovány. Počet tříd poté tvoří poslední rozměr matice masek, tedy 7x48x240x240x10. Kanál, jelikož je pouze jediný, se může zanedbat díky změně tvaru celé matice, která se využívá v architektuře UNET. Data masek jsou z načteného tvaru převedeny na tvar 7x(48·240·240)x10 resp. 7x2764800x10 a dále pomocí funkce `to_categorical` do vicetřídeho tvaru. Jedná se o tvar, kdy je zabezpečena korespondence masky a obrázku. Takový tvar umožňuje neuronové síti stanovit pravděpodobnost každé masky v daném řezu. Z takového formátu se poté převádí zpět reverzní funkcí po provedení predikce na testovacích datech.

---

<sup>1</sup>Jedná se pouze o převod režimu RGB do stupně šedi, data barevnou složku sama o sobě neobsahují.

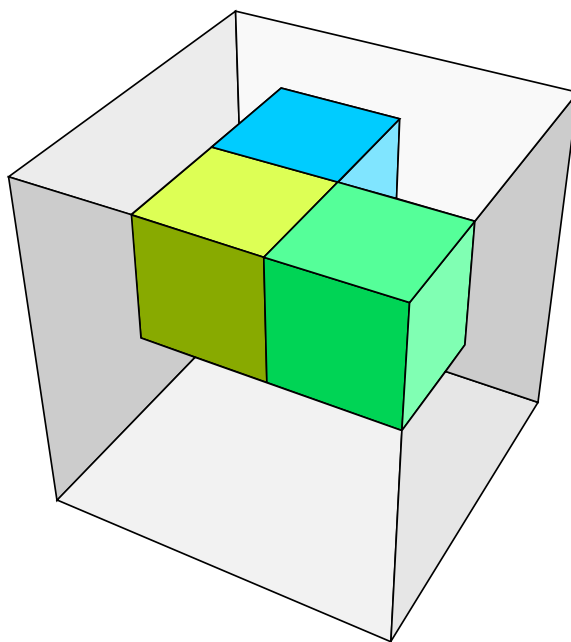
## 3.2 Extrakce 3D obrázku na menší částí

Rozdělení většího trojrozměrného obrázků na menší částí přináší celou řadu výhod jako je větší míra flexibility řešení z hlediska heuristiky nebo různými možnostmi při trénování. A to je také důvod, proč takové řešení bylo implementováno.

Při zpracování velkého datasetu také často vzniká problém s pamětí, protože trénovaný objekt je moc velký na jeho nahrání do ní resp. na takové řešení má daleko větší dopad velikost grafické karty, což je nežádoucí.

Aby se takovému problému předešlo, byl celý sken rozdělen na menší částí (fragments) o dané velikosti. Každý takové menší segment je již možné velice snadno nahrát do paměti. Fragmentů 3D skenu je možné pomocí dávkového zpracování (při trénování) nahrát do paměti hned několik, jelikož jsou samy o sobě dostatečně malé.

Jednou z výhod menších částí pak v takovém případě je, že se data z datasetu mohou nahrát přímo a extrakce na menší částí může probíhat v momentě, než je zahájeno trénování. Další a nejspíše i největší výhodou také je, jak již bylo nastíněno výše, že je možné použít pro extrakci pohyblivé okno tj. jednotlivé fragmenty mozku se mohou v paměti překrývat, čímž je možné ve výsledku dosáhnout přesnější finální predikce.



Obr. 3.1: 3D patche [3]

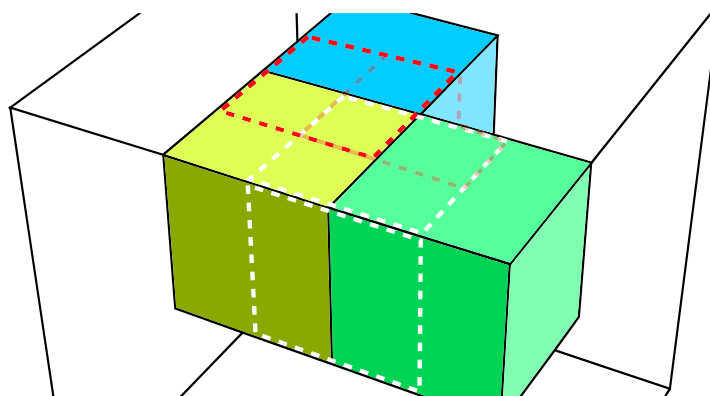
Ukázka rozdělení na menší částí lze vidět na obr. 3.1 výše, kdy velká krychle reprezentuje celý 3D sken hlavy a menší barevné částí zmiňované fragmenty.



### 3.2.1 Rozdělení 3D skenu pomocí posuvného 3D okna

Při extrakci fragmentů ze 3D skenu mozku je možné zvolit určitý překryv, o kolik se budou jednotlivé části překrývat tj. krok a dále velikost 3D okna, která reprezentuje i samotnou velikost cílového fragmentu. Definované 3D okénko se poté posouvá o velikost kroku, což vytváří výsledný průřez. Pokud by např. byl krok posunu 3D okna shodný s jeho velikostí, překryv by byl nulový.<sup>2</sup>

Tímto způsobem tedy díky překryvu vznikne větší množství menších částí skenu, nežli by vůbec mozek obsahoval.<sup>3</sup> Pokud by byly fragmenty extrahovány blokově<sup>4</sup>, kde překryv není žádný a každý segment plynule navazuje na ten následující, tak počet fragmentů již bude shodný.



Obr. 3.2: 3D patche

Na obr. 3.2 je demonstrován 50% překryv fragmentů, kdy barevné 3D kostky reprezentují 3D okno a vyhraničené oblasti bílou a červenou přerušovanou čarou vyznačují posunuté 3D okno o velikost kroku. Posun trojrozměrného okna je možný ve všech třech osách  $x$ ,  $y$  a  $z$ .

## 3.3 Augmentace dat

Hlavní přínos augmentace dat spočívá v tom, že je možné tímto způsobem rozšířit datovou sadu např. v případě, kdy zvolený dataset nedisponuje dostatečným objemem dat. Díky jednoduchým augmentačním technikám pak zůstane zachován fakt, že se síti budou předkládat data, která ještě neviděla a které přispějí k její vyšší obecnosti.

---

<sup>2</sup>Za koncem jednoho fragmentu bude začínat ihned další apod., což samo o sobě odpovídá blokovému zpracování.

<sup>3</sup>V případě blokového zpracování.

<sup>4</sup>V případě, že krok je shodný s velikostí okna.

Mezi další přínosy patří zvýšení tzv. invariance na posuvy apod. V situacích, kdy jsou data pořizována různými technikami pod různými úhly reflektující jednou a tutéž věc, může dojít k tomu, že neuronová síť by taková data, jen trochu odlišná od těch, na které byla trénovaná, již nemusela vůbec rozpoznat.

Neuronová síť má také tendenci k tomu si dataset pamatovat, čímž dochází k přetrénování a obecně ke snížení obecnosti celé sítě.

Augmentovat data lze např. rotací, posuvem, přidáním šumu apod. Díky ní pak bude neuronová síť robustnější a také významně přispěje k její generalizaci.

Další faktor augmentace je ten, že by pozměněná data vůči původním měly pořádk dávat smysl a měly by být pro neuronovou síť přínosem v učení. Jedná se tedy o konkrétní použití dané sítě a od toho se poté odvíjí i postup.

V případě vícetříde segmentace dává smysl otočení hlavy, aby síť byla schopna reagovat na situaci, kdy obrázky nejsou přesně skenovány kolem ose x nebo y. Další použitou augmentační technikou je zvětšení, jelikož skeny mozku nemusí být vždy totožné velikosti a síť by s touto variantou měla počítat.

### **3.3.1 Aplikace augmentace**

V případě augmentace dat jsou možné dva přístupy, jak ji aplikovat. Vstupní data lze augmentovat přímo při zpracování čímž by nedošlo k rozšíření datové sady. Druhá možnost je augmentaci přidat jako rozšíření datasetu, čímž by došlo k tomu, že by neuronová síť byla méně náchylná k přetrénování a zároveň dokázala lépe řešit segmentaci, jelikož deformace, posuvy, šum apod. by nehrál takový vliv při predikci, jelikož by síť i na tyto případy byla trénovaná.

K řešení vícetříde segmentace slouží augmentace zejména k rozšíření trénovací sady, jelikož dataset neobsahuje dostatečně velkou množinu dat pro trénování.

## 3.4 Vytvoření modelu a jeho parametry

Model byl postaven na architektuře Unet. Mezi hlavní důvody, proč byla zvolena tato architektura, patří především to, že dokáže dosahovat dobrých výsledků na menší množině dat a také dále disponuje menší paměťovou náročností.

Samotný proces vytvoření modelu závisí především na vstupních datech, od kterých se odvíjí také parametry modelu. V případě segmentace 3D lékařských dat do více tříd je třeba uvažovat, že síť bude klasifikovat data dle štítků do různých kategorií, které se trénují fakticky separátně. Nejstěžejnější částí modelu vícetříde segmentace je nastavení výstupní vrstvy dle počtu tříd, do kterých se segmentuje a tomu přizpůsobit všechny vstupní data, masky a metriky. Tvar dat, kterými disponuje výstupní vrstva, očekává model poté i u vstupních masek při trénování. Vstupní formát trénovacích dat a masek se neshoduje a také proto tomu bylo nutné přizpůsobit výstupní vrstvu modelu pomocí změny tvaru této vrstvy.

Model má nastavenou aktivační funkci `softmax` na své poslední vrstvě, jelikož se využívá vícetřídní klasifikace. Jako optimalizační *loss* funkci používá `categorical_crossentropy`. Inicializace jader probíhá pomocí normálového rozložení na všech konvolučních 3D vrstvách.

Korekce výstupu ke vstupním datům během trénování dochází tak, že `categorical_crossentropy` vypočítává pravděpodobnost jednotlivých tříd pro každý pixel úpravou pole o velikosti shodné s počtem segmentovaných tříd, kde každá hodnota pole reprezentuje pravděpodobnost dané třídy, která je dána indexem hodnoty pole.

## 3.5 Rekonstrukce dat

V momentě ukončení trénování je na řadě predikce, která probíhala ne jediném 3D obrázku resp. více obrázcích vytvořených z původního 3D skenu pomocí augmentace. Výsledná výstupní predikce masek, která je výstupem modelu, se dále musí ještě zpracovat. Jelikož `categorical_crossentropy` vypočítává pravděpodobnost jednotlivých tříd pro jednotlivé body obrazu, je nutné vybrat nejpravděpodobnější třídu, a tu pak aplikovat na daný pixel, čímž dojde k vysegmentování pixelu do dané třídy. Každý pixel disponuje polem o velikosti  $10^5$  a hodnota, která nabývá nejvyšší hodnoty v tomto poli je predikovaná třída pixelu. Zařazení obrazového bodu do třídy následně probíhá tak, že index vítězné třídy se přiřadí hodnotě pixelu v momentě zpracování predikovaných masek.

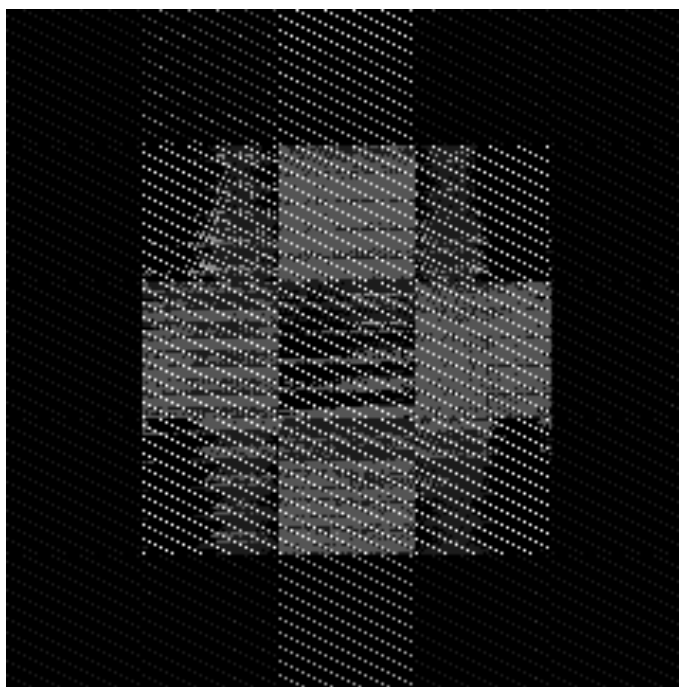
Pokud predikce proběhne nad nefragmentovanými řezy resp. skenem, který nebyl pomocí 3D okna fragmentován, tak výstup odpovídá stejnému formátu, jaký do

---

<sup>5</sup>Velikost odpovídá počtu segmentovaných tříd.

predikce byl nahrán. Situace je tedy složitější v případech, kdy bylo 3D okno použito pro zpracování dat.

Predikce dat zde probíhá pomocí metody `fit_predict`, která má na vstupu jednotlivé fragmenty 3D skenu a pro každý fragment predikuje výslednou masku. Následné zpracování obrázku tedy probíhá tak, že se nejprve vypočítá pomocí metody `argmax` výsledná třída segmentace pro všechny řezy 3D fragmentu, díky čemuž je již možné jednotlivé řezy složit do výsledného fragmentu. Poslední krok zpracování 3D skenu z predikce spočívá ve složení jednotlivých fragmentů, kterým byly vypočítány klasifikace.



Obr. 3.3: Ukázka sestavení jednoho řezu z fragmentů.

Na obr. 3.3 je demonstrováno skládání 3D skenu z jednotlivých fragmentů po natrénování cca 2 epoch. Z obrázku jsou ještě patrné jednotlivé bloky, z kterých byl daný řez složen. Postupem trénování se vzrůstající přesností dochází k zahlazování jednotlivých bloků a zvyšování detailů až do té míry, že fragmentace již není patrná. Na obrázků již je i patrný obrys výsledné vícetřídě segmentace mozku.

## 4 Výsledky a jejich vyhodnocení

Následující kapitola diplomové práce se zabývá procesem trénování na vstupních datech a generováním výsledné predikce (cílové vícetříde segmentace obrazu). Výslednou predikci neuronové konvoluční sítě dále vyhodnocuje podle doporučených metrik a prezentuje dosažené výsledky.

### 4.1 Metriky pro hodnocení segmentace

Výsledek segmentace je třeba nějakým způsobem ohodnotit resp. stanovit přesnost mezi vysegmentovaným objektem a tím referenčním. Vyhodnocení práce následuje metriky, které byly též použity při hodnocení výsledků soutěže nad daným datasetem. Metriky berou v úvahu všechny segmentované třídy a stanovují výsledný koeficient pro každou třídu zvlášť tak, aby byla jasná přesnost každé z nich.

#### 4.1.1 Dice koeficient

Jedná se o statistickou metodu, která slouží ke stanovení podobnosti dvojice vzorků. Rovnice pro výpočet koeficientu je definována jako 4.1:

$$DSC = \frac{2 \cdot |X \cap Y|}{|X| + |Y|} \quad (4.1)$$

Množiny  $Y$  a  $X$  reprezentují zkoumané vzorky resp. predikovanou a referenční segmentaci mozku.

#### 4.1.2 Hausdorffova vzdálenost

Metrika určuje maximální vzdálenost bodu testovaného vzorku  $\mathbf{A}$  k nejbližšímu bodu vzorku  $\mathbf{B}$ . Ke stanovení koeficientu využívá dvojici funkcí tj. nejprve funkci pro kalkulaci vnitřních vzdáleností a poté operaci k vyhodnocení vypočtených hodnot tj.  $\max()$ . Obecně by se hausdorffova vzdálenost dala popsat následujícím vztahem:

$$H(A, B) = \max\{h(A, B), h(B, A)\} \quad (4.2)$$

kde  $h(A, B)$  a  $h(B, A)$  reprezentují vnitřní funkci pro určení vzdálenosti bodů. Zároveň platí, že ve většině případů  $h(A, B) \neq h(B, A)$ . [45]

### 4.1.3 Snímková podobnost

Poslední z metrik je snímková podobnost, která je počítána na základě vztahu:

$$VS = 1 - \frac{|V_a - V_b|}{V_a + V_b} \quad (4.3)$$

podobně jako u předešlých metrik množina  $V_a$  symbolizuje množinu referenční segmentace a dále  $V_b$  hodnoty predikce.

## 4.2 Náročnost výpočtu a parametry trénování

Trénování neuronové sítě probíhalo na serveru *gravy*. K dispozici byly grafické karty NVIDIA Titan XP a NVIDIA GeForce GTX 1080 Ti. K trénování sítě byly využívány obě tyto grafické karty.

Trénování neuronové konvoluční sítě pomocí fragmentů zabralo 61 hodin. Výpočet nad celými skeny trval 28 hodin. Znatelný rozdíl v době trénování byl způsoben především kvůli překryvu, který v případě trénování sítě po menších částech činil 50%. Velikost jednoho fragmentu tedy činil (48, 48, 48). Vlivem překryvu vzniklo na každých 5 fragmentů 4 překryvné. Jediný 3D sken se poté skládal z 81 částí. Dávkování 3D skenu bylo prováděno sekvenčně a každá dávka fragmentů byla při trénování po každé epoše zamíchána.

Datová sada byla zvětšena pomocí augmentačních metod otočení a přiblížení celkem 10x. Metoda augmentace otočení byla aplikována v následujících variantách 7°, -7°, 12°, -12°, 17°, -17°, kde záporné/kladné hodnoty udávají směr otočení. Dále bylo aplikováno přiblížení snímků resp. zvětšení a to na hladiny: 105%, 110%, 117% a 121% původní velikosti 3D skenu.

Optimalizace byla prováděna pomocí metody Adam s učícím koeficientem  $1e-5$  a v obou případech trénování byla použita metrika přesnosti *accuracy*.

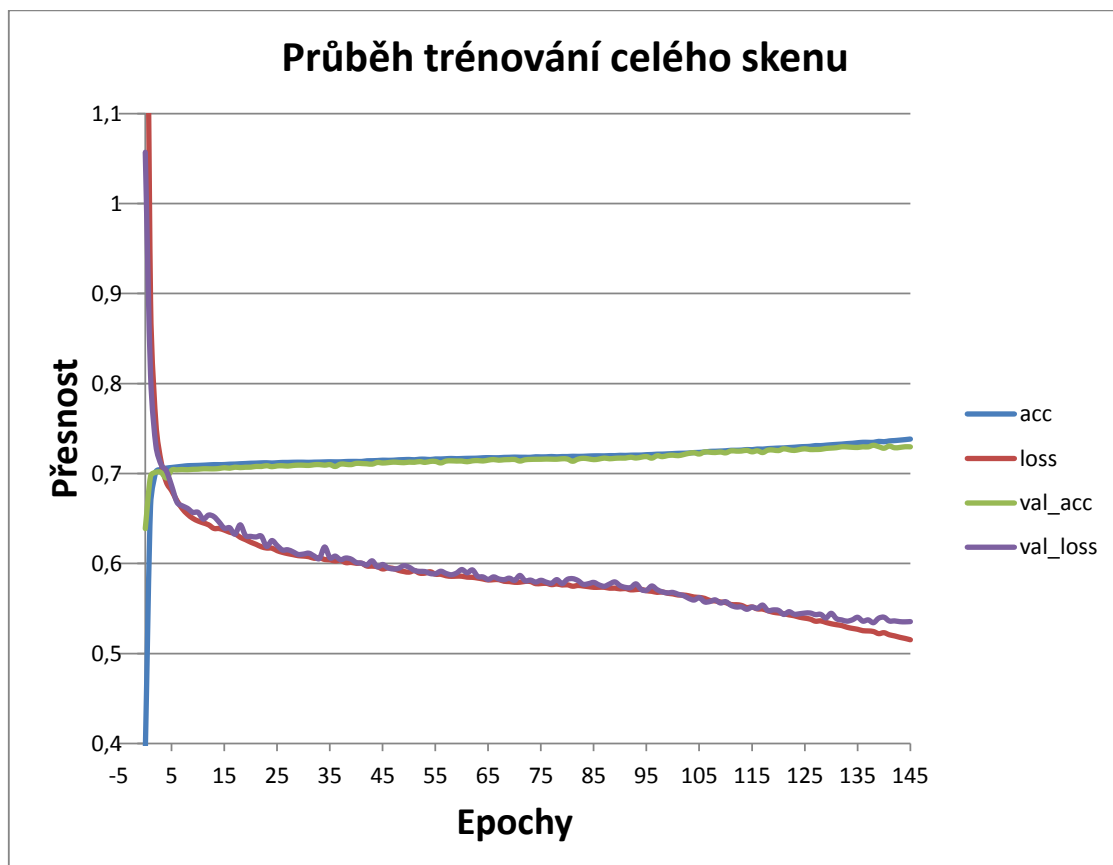
## 4.3 Průběh trénování neuronové sítě

Jak již bylo nastíněno výše, byly vyzkoušeny dvě možnosti provedení celé vícetřídě segmentace, které se liší způsobem přístupu k datům zvoleného datasetu, což se poté promítá i v dalších částech zpracování jako např. parametry modelu nebo zacházení z daty po provedení trénování a stanovení výsledné predikce neuronovou sítí. Cílem tedy je provést vícetřídou segmentaci a ověřit, který z přístupů může produkovat lepší výsledky.

### 4.3.1 Trénování pomocí celých 3D skenů

Následující část demonstruje trénování nad celými 3D skeny tak, jak byly dodávány v datasetu. Metriky `acc` a `loss` udávají hodnoty nad trénovací množinou a `val_acc` a `val_loss`<sup>1</sup> nad validační.

Vztah růstu přesnosti a vývoj chyby v průběhu jednotlivých epoch lze pozorovat na obr. 4.1.



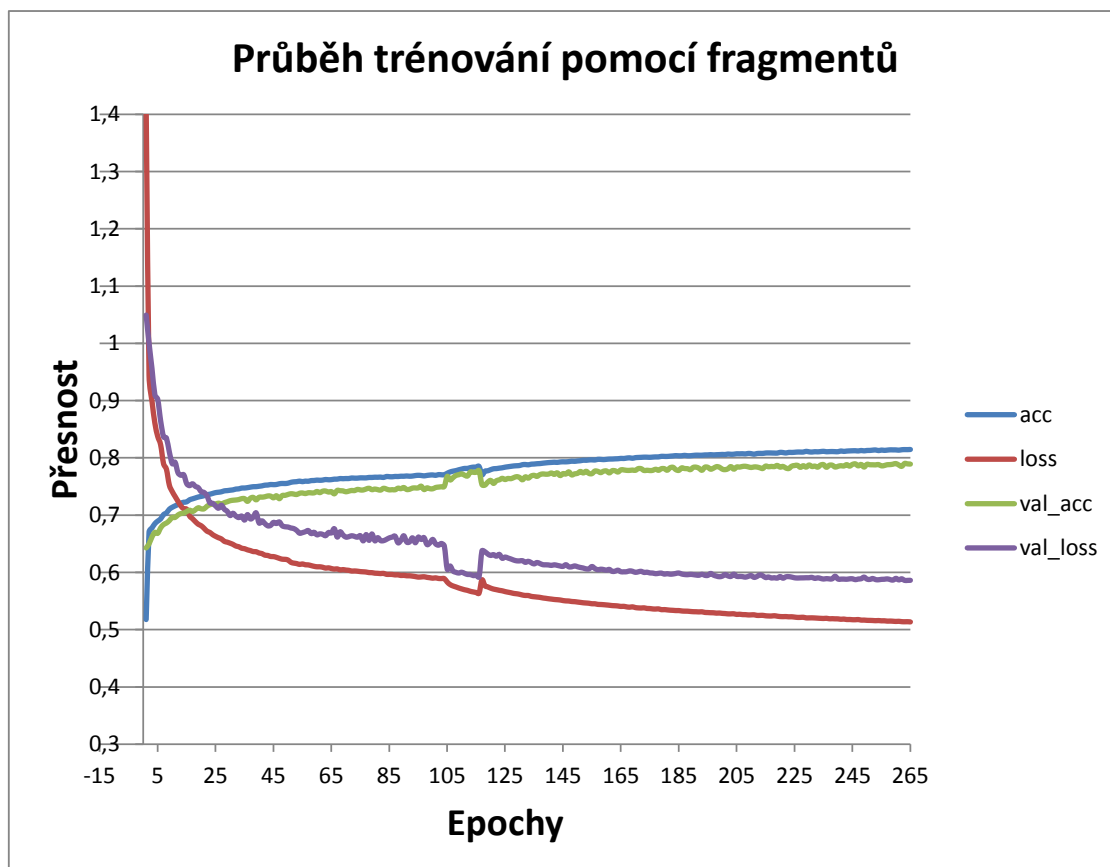
Obr. 4.1: Průběh trénování po celých skenech.

Z grafu lze pozorovat, že trénování sítě skončilo cca po 145 epochách. Konec trénování způsobil především fakt, že se již poslední epochy sítě nezlepšovala a začalo tzv. přetrénování. V této chvíli byl tedy výpočet ukončen. Tuto skutečnost lze pozorovat i přímo z grafu trénování, kde začíná být znatelný rozdíl v křivce `loss` a `val_loss`, kdy `loss` jde dolů, což znamená, že sítě se zlepšovala jen na trénovacích datech ale predikce byla buď horší nebo stagnovala.

<sup>1</sup>V našem případě nejdůležitější metrika.

### 4.3.2 Trénování pomocí fragmentů

Situace v případě trénování pomocí fragmentů je dost odlišná. Vztah růstu přesnosti a vývoj chyby v průběhu jednotlivých epoch při trénování pomocí fragmentů vyobrazuje graf 4.2.



Obr. 4.2: Průběh trénování.

Z průběhu je znatelná pozvolnější konvergence. Celé trénování proběhlo na 265 epoch, přičemž jedna epocha trvala cca 14 minut. Po celém průběhu se síť zlepšovala a dosáhla výsledné přesnosti cca 0.82 v případě metriky **accuracy**. Proces trénování byl ukončen po příliš malém přírůstku delta parametru resp. míry zlepšení mezi dvěma cykly trénování.



## 4.4 Výsledky segmentace

Po provedení predikce byly na výsledné skeny masek aplikovány metriky dice-koeficient, hausdorffova vzdálenost a snímková podobnost. U výsledných masek bylo ve všech případech ignorováno pozadí a metriky vyhodnocovaly třídy 1-8 viz. tabulka 2.1. Stejně tak jako pozadí (třída 0) byla při měření zanedbána i třída 9 resp. infarkt. Důvodem, proč byla tato třída vyloučena ze soutěže je ten, že v datasetu nebyla zastoupena v dostatečné míře. Jelikož bylo pozadí při vyhodnocení zanedbáno, tak i z výsledné predikce bylo zcela odstraněno post-zpracováním predikovaných masek.

Dice koeficient nabývá hodnot od 0-1 přičemž nejlepší výsledek je 1. Stejně tak v případě snímkové podobnosti. Odlišný interval hodnot má až metrika hausdorff, u které platí, že čím méně, tím lépe. Nejlepší možnou hodnotou v případě této metriky je tedy 0.

### 4.4.1 Přesnost trénování po celých 3D skenech

Dle výsledků tabulky 4.1 je zřejmé, že některé třídy mají nízkou přesnost. Třída CSF (Cerebrospinální tekutina resp. mozkomíšní mok - *Cerebrospinal fluid*) a BG (Bazální ganglia - *Basal ganglia*) vykazují velmi nízké hodnoty.

#	CGM	BG	WM	WML	CFS	V	C	BS
DSC	0.15	0.00	0.35	0.00	0.05	0.09	0.11	0
H	13.85	61.00	9.50	30.44	78.03	50.68	58.73	–
VS	0.44	0.65	0.91	0.92	0.16	0.33	0.55	0

Tab. 4.1: Výsledky trénování pomocí celých skenů.

Třidu BS (Mozkový kmen - *Brain stem*) se vysegmentovat nepodařilo, tudíž DSC (Dice koeficient - *Dice coefficient*) a VS (Snímková podobnost - *Volume similarity*) jsou nulové, podobně pak i metrika H (Hausdorffova vzdálenost - *Hausdorff distance*), která je pomyslně rovna nekonečnu, čímž potvrzuje výsledek předchozí dvojice metrik.

Nejlepších výsledků pak segmentace dosahuje u tříd WM (Bílá hmota - *White matter*) a CGM (Kortikální šedá hmota - *Cortical gray matter*).

#### 4.4.2 Přesnost trénování po fragmentech

Následující případ opět vykazuje zcela odlišný výsledek v porovnání s tím předešlým. Z hodnot v tabulce 4.2 je patrné, že nejenže došlo k vysegmentování všech tříd, ale některé se podařilo vysegmentovat s relativně vysokou přesností. Nejméně se pak dařilo vysegmentovat třídu WML (Léze bílé hmoty - *White matter lesions*) a BG (Bazální ganglia - *Basal ganglia*).

#	CGM	BG	WM	WML	CFS	V	C	BS
DSC	0.54	0.24	0.62	0.17	0.67	0.79	0.86	0.87
H	2.24	4.69	4.69	16.12	3.16	7.00	4.00	3.74
VS	0.90	0.67	0.95	0.24	0.96	0.90	0.96	0.93

Tab. 4.2: Výsledky trénování pomocí fragmentů.

Naopak BS (Mozkový kmen - *Brain stem*), V (Mozkové komory - *Ventricles*) a C (Mozeček - *Cerebellum*) dosahují nejvyšších přesností. V zásadě by se dle výsledku trénování pomocí fragmentů dalo říci, že přesnost tříd je přímo závislá na procentuálním zastoupení dané třídy v mozku. Pokud tedy má třída vyšší zastoupení v mozku, je to dobrý předpoklad pro úspěšnější predikci, což tedy i potvrzuje hlavní důvod, proč byla třída infarktu vynechána.



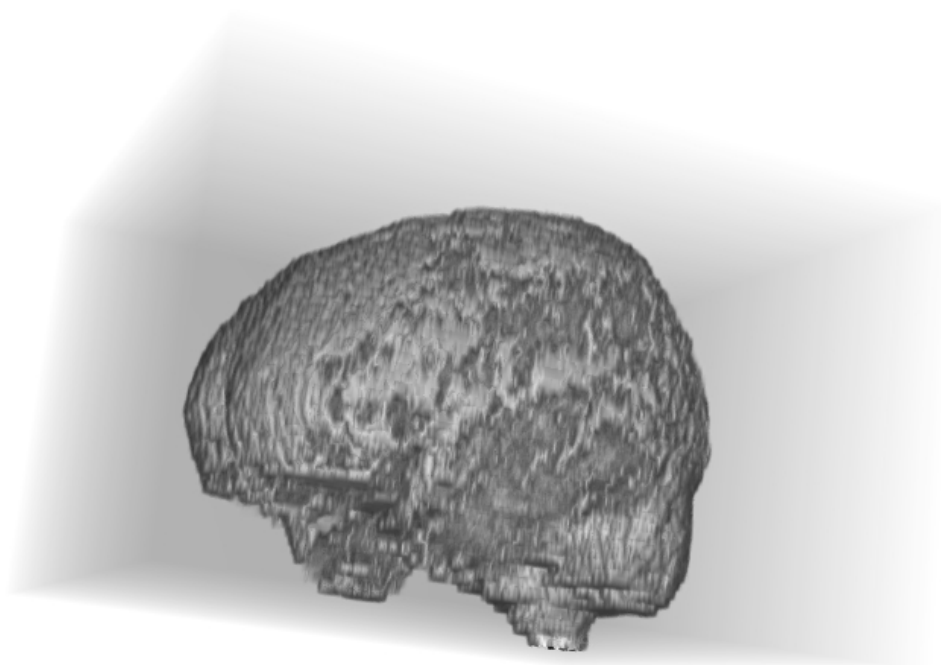
Obr. 4.3: Řez výsledné vícetříde segmentace.

Na obr. 4.3 se nachází 26 řez transverzální roviny z predikovaného 3D skenu. Kromě segmentovaných tříd jsou v obrázku zjevné i nežádoucí artefakty, které vznikly během překryvu při dekonvoluci.

#### 4.4.3 Výsledná 3D segmentace

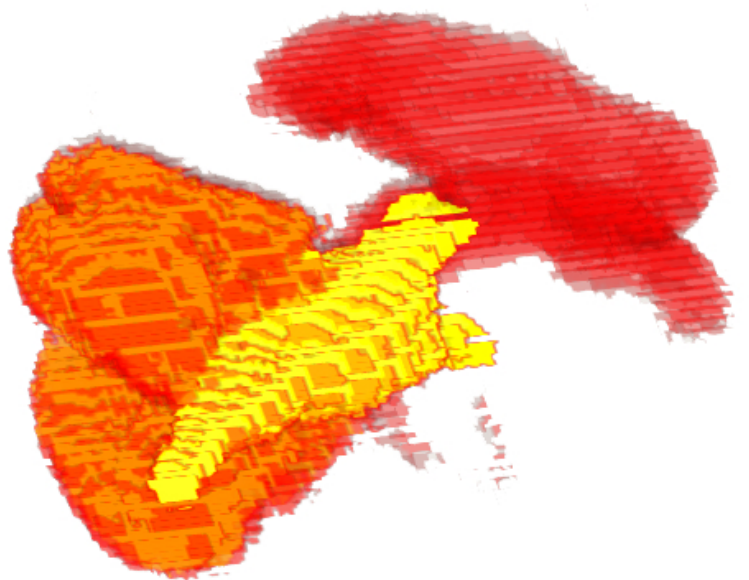
Každá třída segmentace je reprezentována jinou intenzitou stupně šedi. Referenční masky nabývají intenzity dle třídy tj. 0-9, což je pro vizualizace nedostačující a také proto byla pro tyto účely zvýšena intenzita jednotlivých úrovní tříd tak, aby byl výsledek predikce viditelný.

Na obrázku 4.4 lze vidět výslednou segmentaci celého mozku, který je poskládán z jednotlivých tříd.



Obr. 4.4: Výsledná 3D segmentace celého mozku.

Segmentaci několika tříd poté demonstruje obr. 4.5. Segmentované třídy jsou zvýrazněny barevně, kdy třída C (Mozeček - *Cerebellum*) je vyznačen oranžovou barvou, dále třída V (Mozkové komory - *Ventricles*), která je zvýrazněna červenou barvou a žlutou barvou je zvýrazněn BS (Mozkový kmen - *Brain stem*).



Obr. 4.5: Výsledná 3D segmentace mozku do více tříd.

## 4.5 Vyhodnocení výsledků a možné pokračování práce

Z naměřených hodnot a provedených výpočtů jasně vyplynulo, že metodou rozložení mozku na menší části lze provést úspěšnější vícetřídní segmentaci, nežli v případě trénování po celých 3D skenech. Daří se tak díky menším částem mozku lépe reflektovat změny, aby segmentace dosahovala ve výsledku lepší predikce.

Díky této metodě se dařilo na základě malých fragmentů lépe upravovat váhy modelu, aby byla celá vícetřídní segmentace přesnější. Z toho důvodu by bylo velice zajímavé tento přístup dále rozvinout a pokusit se dále optimalizovat pro vícetřídní segmentaci. Jednou ze zajímavých možností jak pokračovat by byl méně deterministický způsob dávkování trénovacích dat do modelu např. náhodným volením dávek, což by více přispělo ke generalizaci sítě. Dalším možným způsobem by mohlo být trénování s menší velikostí fragmentů a porovnat poté jejich výsledek navzájem např. o velikosti (8,12,12) a (16,16,16). Model v takovém případě může být daleko menší a vzniká velké množství dalších množných kombinací, jak segmentaci provést.

Mezi další možnosti, jak se lze pokusit výsledky vícetřídní segmentace zlepšit, je použití jiných architektur konvolučních neuronových sítí, jako je například již zmiňovaný ResNet, VGG v kapitole 2.2.5 apod. Popřípadě vyzkoušení tří-kanálové extrakce příznaků pro každý typ skenu pomocí samostatného enkodéru v kombinaci s jedním dekodérem. Lze také vyzkoušet větší adaptaci zvoleného modelu přesně na daný dataset či jiné modifikace.

Díky fragmentaci 3D skenu je možné pracovat ještě s daleko většími obrázky, jelikož se pak sken může díky fragmentům rozdělit na menší části a ty zpracovávat samostatně. Z toho také plyne další možnost, jak dosáhnout lepších výsledků. V případech, kdy je obrázek malý pro vícetřídní segmentaci, je možné obrázky zvětšit na daleko větší velikost, čímž by se zvedlo zastoupení detailů vůči celému obrázku a tedy i síť bude schopna lepší predikce.

## 5 Závěr

Cílem diplomové práce byla vícetřídní segmentace 3D lékařských dat pomocí hlubokého učení. Segmentace byla prováděna nad datasetem *mrbrains18*, který byl publikován jako výzva ve třetím čtvrtletí 2018 a publikován na konferenci MICCAI (*Medical Image Computing and Computer Assisted Intervention*).

Diplomová práce je dělena do čtyř základních kapitol. První kapitola se zabývá seznámením s neuronovými sítěmi, na kterých stojí i základy této práce. Jsou zde popsány principy neuronových sítí, jejich vrstvy a dělení. Kapitola umožňuje se rychle zorientovat v neuronových sítích a umožňuje jejich snadné pochopení. Další kapitolou je analýza problému, která se zaměřuje blíže na segmentační metody, techniky a vývojové prostředí pro práci s konvolučními neuronovými sítěmi. Dále se zabývá popisem dat, které jsou v práci přímo zpracovávány pro její účely. Předposlední část práce se zabývá postupy zpracování dat, sestavením modelu a implementací jako takovou. Poslední část práce je věnována vyhodnocení výstupních masek spolu s dosaženou přesností.

Ve vícetřídní segmentaci dramaticky rostou nároky na úspěšnou segmentaci. Ve srovnání s binární segmentací, kdy má neuronová síť pouze rozhodnout o jednoduché segmentaci tj. max jedné třídy, u vícetřídní se náročnost násobí. Navzdory podmínkám, se kterými byla vícetřídní segmentace prováděna, byla vykonána úspěšně. Síť se potýkala zejména s nedostatkem dat, jelikož dataset se skládal pouze ze sedmi skenů, kde každý měl sice trojici skenů v různých režimech, ale i tak neuronová síť potřebovala pro trénování větší dataset. Z toho důvodu byla trénovací sada rozšířena pomocí augmentačních technik rotace a zvětšení.

Výsledkem práce je 3D segmentace lékařských dat do více tříd. Vstupními daty jsou výstupy z magnetické rezonance, které jsou poté segmentovány do různých tříd pomocí neuronových sítí. Postup zpracování vstupních dat byl zvolen pomocí exportu jednotlivých řezů hlavy, na kterých se pak celá neuronová síť učila. Model neuronové sítě následuje architekturu U-net, která byla modifikována tak, aby byla schopna realizovat vícetřídní rozeznávání.

Hlavním přínosem práce je řešení problematiky 3D segmentace do více tříd a návrhem dílčích částí tak, aby tento problém mohla síť efektivně řešit. Práce nabízí hned dva možné přístupy, jak vícetřídní segmentaci provést. Dalším přínosem práce je zpracování celého úvodu do neuronových sítí s podrobným popisem vnitřních procesů, díky kterým usnadňuje jejich pochopení. Mezi další přínosy patří detailní přehled segmentačních metod, které se v současné době při segmentaci využívají. Výsledek této práce by se dal použít jako základ pro další pokračování v 3D segmentaci např. při zvyšování přesnosti výsledků nebo zvětšení počtu klasifikačních tříd apod.

# Literatura

- [1] US National Cancer Institute's Surveillance and redrawn by Dhp1080. Anatomy and physiology - neurone. [online], 3 2007. URL: <<https://commons.wikimedia.org/wiki/File:Neuron.svg>>.
- [2] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). URL: <http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a>.
- [3] Takashi Yoshino, Toyo University. Rubik's 4-cube. [online], 12 2017. URL: <[https://www.mathematica-journal.com/data/uploads/2017/12/Yoshino\\_Output\\_8.gif](https://www.mathematica-journal.com/data/uploads/2017/12/Yoshino_Output_8.gif)>.
- [4] MRBrainS18. Grand challenge on mr brain segmentation at miccai 2018. [online], 9 2018. URL: <<http://mrbrains18.isi.uu.nl/data/>>.
- [5] PhD. doc. RNDr. PaedDr. Eva Volná. *Neuronové sítě*. Ostravská univerzita v Ostravě, Ostrava, 1 edition, 2013.
- [6] PhD. RNDr. PaedDr. Eva Volná. *Neuronové sítě 1*. Ostravská univerzita v Ostravě, Ostrava, 2 edition, 2008.
- [7] Lucie Görlichová. Umělé neuronové sítě v lékařské diagnostice. Master's thesis, MASARYKOVA UNIVERZITA V BRNĚ, 2006.
- [8] CSc. prof. Ing. Ivo Vondrák. *Umělá inteligence a neuronové sítě*. Vysoká škola báňská - Technická univerzita, Ostrava, 1 edition, 1998.
- [9] Citizendium. Artificial neural network. [online], 11 2009. URL: <[http://en.citizendium.org/wiki/Artificial\\_neural\\_network](http://en.citizendium.org/wiki/Artificial_neural_network)>.
- [10] MUDR. LADISLAV POLÁCH. Neurony, synapse a neurotransmitery. [online], 8 2015. URL: <<http://www.upsychiatra.cz/neurony-synapse-a-neurotransmitery/>>.
- [11] Ivo Šnábl. Úvod do neuronových sítí - biologická analogie. [online], 11 2018. URL: <<http://portal.matematickabiologie.cz/index.php?pg=analiza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron>>.

- [12] Mendelova univerzita v Brně. Neuronové sítě. [online], 11 2018. URL: <[https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=21471](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471)>.
- [13] FIT ČVUT. Umělé neuronové sítě a neuroevoluce. [online], 11 2018. URL: <<https://users.fit.cvut.cz/~rehorto2/otevrena-fakulta/neural-networks.html>>.
- [14] Yann LeCun. Lenet-5 convolutional neural networks. [online], 11 2018. URL: <<http://yann.lecun.com/exdb/lenet/>>.
- [15] Ing. Vítězslav Beran Ing. Michal Španěl. Obrazové segmentační techniky - přehled existujících metod. [online], 1 2006. URL: <<http://www.fit.vutbr.cz/~spanel/segmentace/>>.
- [16] Paul-Louis Pröve. An Introduction to different Types of Convolutions in Deep Learning. [online], 7 2017. URL: <<https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d>>.
- [17] Ryuhei Hamaguchi, Aito Fujita, Keisuke Nemoto, Tomoyuki Imaizumi, Shuhei Hikosaka. Effective Use of Dilated Convolutions for Segmenting Small Object Instances in Remote Sensing Imagery. [online], 9 2017. URL: <<https://arxiv.org/ftp/arxiv/papers/1709/1709.00179.pdf>>.
- [18] LISA lab. Convolutional neural networks (lenet). [online], 1 2018. URL: <<http://www.deeplearning.net/tutorial/lenet.html>>.
- [19] Ujjwal Karn. An intuitive explanation of convolutional neural networks. [online], 8 2016. URL: <<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>>.
- [20] Fei-Fei Li & Justin Johnson & Serena Yeung, Stanford University. Training neural networks, part i. [online], 4 2017. URL: <[http://cs231n.stanford.edu/slides/2017/cs231n\\_2017\\_lecture6.pdf](http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture6.pdf)>.
- [21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015. URL: <http://dl.acm.org/citation.cfm?id=3045118.3045167>.
- [22] Federico Peccia. Batch normalization: theory and how to use it with Tensorflow. [online], 9 2018. URL: <<https://towardsdatascience.com/batch-normalization-theory-and-how-to-use-it-with-tensorflow-1892ca0173ad>>.



- [23] François Chollet. Keras: The python deep learning library. [online], 3 2015. URL: <<https://keras.io/>>.
- [24] Computer Science Department, Stanford University. Neural networks. [online], 6 2013. URL: <[http://ufldl.stanford.edu/wiki/index.php/Neural\\_Networks](http://ufldl.stanford.edu/wiki/index.php/Neural_Networks)>.
- [25] Medium Uniqtech Co. Understand the softmax function in minutes. [online], 1 2018. URL: <<https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d>>.
- [26] Juan Miguel Valverde. Activation functions in deep learning. [online], 1 2018. URL: <<http://laid.delanover.com/activation-functions-in-deep-learning-sigmoid-relu-lrelu-prelu-rrelu-elu-softmax/>>.
- [27] Stanislav Straka. Segmentace obrazu. Master's thesis, MASARYKOVA UNIVERZITA - FAKULTA INFORMATIKY, 2009.
- [28] Milan ŠONKA a Václav HLAVÁČ. *Počítačové vidění*. Grada, Praha, 1 edition, 1992.
- [29] Ilona Janáková. Segmentace. [online], 1 2018. URL: <[http://midas.uamt.feec.vutbr.cz/POV/Lectures/05\\_Segmentace.pdf](http://midas.uamt.feec.vutbr.cz/POV/Lectures/05_Segmentace.pdf)>.
- [30] Ruye Wang. Edge detection. [online], 10 2016. URL: <<http://fourier.eng.hmc.edu/e161/lectures/gradient/node6.html>>.
- [31] Bc. ONDŘEJ MIKUŠ. Detektor objektů s využitím vlnkové transformace. Master's thesis, VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ - FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH TECHNOLOGIÍ, 2009.
- [32] MINH TUAN PHAM. Segmentace mri dat s využitím metody narůstání oblastí. Master's thesis, VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ - FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH TECHNOLOGIÍ, 2012.
- [33] Ilona Janáková. Detekce geometrických primitiv. [online], 1 2018. URL: <[http://midas.uamt.feec.vutbr.cz/POV/Lectures/06\\_Detekce\\_geometrickych\\_primitiv.pdf](http://midas.uamt.feec.vutbr.cz/POV/Lectures/06_Detekce_geometrickych_primitiv.pdf)>.
- [34] Bc. PAVEL SVOBODA. Vyhledávání osob ve fotografii. Master's thesis, VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ - FAKULTA INFORMAČNÍCH TECHNOLOGIÍ, 2009.

- [35] Siddharth Das. CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and . [online], 10 2017. URL: <<https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>>.
- [36] SUNITA NAYAK. Understanding AlexNet. [online], 6 2018. URL: <<https://www.learnopencv.com/understanding-alexnet/>>.
- [37] Christian Szegedy Wei Liu Yangqing Jia Pierre Sermanet Scott Reed. Going Deeper with Convolutions. [online], 6 2018. URL: <<https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf>>.
- [38] Sik-Ho Tsang. Review: GoogLeNet (Inception v1)— Winner of ILSVRC 2014 (Image Classification). [online], 8 2018. URL: <<https://medium.com/coinmonks/paper-review-of-googlenet-inception-v1-winner-of-ilsvrc-2014-image-classification-c2b3565a64e7>>.
- [39] Karen Simonyan & Andrew Zisserman, Visual Geometry Group, Department of Engineering Science, University of Oxford . VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. [online], 8 2018. URL: <<https://arxiv.org/pdf/1409.1556.pdf>>.
- [40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Microsoft Research. Deep Residual Learning for Image Recognition. [online], 12 2015. URL: <<https://arxiv.org/pdf/1512.03385.pdf>>.
- [41] LISA lab. U-net. [online], 1 2015. URL: <<http://deeplearning.net/tutorial/unet.html>>.
- [42] Google Brain Team. An open source machine learning framework for everyone. [online], 11 2015. URL: <<https://www.tensorflow.org/>>.
- [43] Eli Bressert. *SciPy and NumPy*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 1 edition, 2013.
- [44] Stéfan van der Walt. Image processing in python. [online], 8 2009. URL: <<https://scikit-image.org/>>.
- [45] Normand Grégoire and Mikael Bouillot, McGill University, Montreal, Quebec, CANADA. Hausdorff distance between convex polygons. [online], 9 1998. URL: <<http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html>>.

## Seznam symbolů, veličin a zkratek

<b>CT</b>	Výpočetní tomografie - <i>Computed Tomography</i>
<b>MRI</b>	Magnetická rezonance - <i>Magnetic resonance imaging</i>
<b>CNN</b>	Konvoluční neuronová síť - <i>Convolutional neural network</i>
<b>ReLU</b>	<i>Rectified Linear Unit</i>
<b>API</b>	Rozhraní pro programování aplikací - <i>Application Programming Interface</i>
<b>FCN</b>	Plně konvoluční síť - <i>Fully Convolutional Network</i>
<b>NIFTI</b>	IT iniciativa pro neurovizualizaci - <i>Neuroimaging Informatics Technology Initiative</i>
<b>CSF</b>	Cerebrospinální tekutina resp. mozkomíšní mok - <i>Cerebrospinal fluid</i>
<b>MICCAI</b>	<i>Medical Image Computing and Computer Assisted Intervention</i>
<b>CGM</b>	Kortikální šedá hmota - <i>Cortical gray matter</i>
<b>BG</b>	Bazální ganglia - <i>Basal ganglia</i>
<b>WM</b>	Bílá hmota - <i>White matter</i>
<b>WML</b>	Léze bílé hmoty - <i>White matter lesions</i>
<b>V</b>	Mozkové komory - <i>Ventricles</i>
<b>C</b>	Mozeček - <i>Cerebellum</i>
<b>BS</b>	Mozkový kmen - <i>Brain stem</i>
<b>I</b>	Infarkt - <i>Infarction</i>
<b>H</b>	Hausdorffova vzdálenost - <i>Hausdorff distance</i>
<b>DSC</b>	Dice koeficient - <i>Dice coefficient</i>
<b>VS</b>	Snímková podobnost - <i>Volume similarity</i>

# Seznam příloh

A Obsah přiloženého CD

68

## A Obsah přiloženého CD

Přiložený disk obsahuje zdrojové soubory diplomové práce, včetně obrázků a přeložené verze pdf. Dále zdrojové soubory pro zpracování, trénování a vyhodnocení daného datasetu.

```
/ ..... kořenový adresář přiloženého CD
├── doc ..... zdrojové soubory diplomové práce (.tex)
├── dataset ..... dataset
├── data ..... zpracovaná data z datasetu
│   ├── train ..... trénovací sada
│   ├── test ..... testovací sada
│   ├── masks ..... sada masek k trénovacím datům
│   └── predict ..... výsledek predikce
├── dataset.py ..... soubor pro zpracování datasetu do png
├── data.py ..... skript k vytvoření numpy matic
├── train.py ..... soubor pro trénování sítě
├── models.py ..... modely konvolučních neuronových sítí
├── predict.py ..... skript pro vytvoření predikce nad modelem
├── eval.py ..... vizualizace výsledků a vyhodnocení přesnosti
├── tools.py ..... knihovna funkcí
└── generator.py ..... knihovna tříd
```